MARGINAL CHECKING AND DIAGNOSTIC PROCEDURES FOR THE WISC

By,

AILAN KEITH SCIDMORE

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of

DOCTOR OF PHILOSOPHY

at the

UNIVERSITY OF WISCONSIN

1957

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF ROUTINES

CHAPTER I

INTRODUCTION

A. Reasons For This Study

Following the design and preliminary construction of a digital computer, there still remain a few tasks before the machine can provide useful computing time. The first of these might be called testing or debugging and has as its major objective the determination that the machine is logically sound. In the WISC, an integrally synchronized computer, timing is very critical since parts of four orders are carried out simultaneously. To complicate matters further, the timing pattern is dependent not only upon the sequence of orders, but also upon the locations from which operands are being taken and the location where the result is to be written. Hence, many combinations of orders and locations of operands and results must be considered. The errors found in the logical design of the WISC have mostly been of this type; however, a large number of construction errors have also been found. In a machine where so many have contributed to the design and construction over such a long period of time, it was inevitable that such logical oversights and construction errors would occur.

In order to accomplish this debugging it is necessary to devise a number of routines to allow observation of machine operation under a specific set of conditions. Although a number of these routines will never be used again; variations of parts of these routines may find application later in preventive maintenance and error diagnosis.

The second task to be performed is to bring the machine to operating condition. After debugging, the machine is known to be logically sound, but it may be marginally reliable; the problem here is one of increasing

the reliability of the machine so that it may be made available for computation, demonstration, etc. To accomplish this the machine operation is tested under severe operating conditions by causing power supply voltages, heater voltages, signal levels, etc., to vary from their normal values. Minor changes in circuit design, replacement of faulty components and occasionally minor logical changes are generally necessary to produce a larger margin of reliability. Here again some special routines are used to test the machine while the operating conditions are varied; these routines will form the basis of the preventive maintenance routines to be used later.

After the machine has been brought to the operating stage, it is desirable that it remain in this condition a large percent of the time. The problems here, then, are of three types:

1. Present Reliability

Some method of determining the reliability of the computer while in use must be provided. There are at present two approaches to this problem of assuring the operator that the computer is functioning correctly. One approach is to design and construct the computer in such a fashion that it automatically checks its own operations, while the other is to provide such checks by means of programming.

An extreme example of the first or self-checking school of thought was the BINAC which consisted of two identical and complete computers checking each other at all times.[1] In the Central Computer in the Univac System the duplicated and checking equipment is about thirty percent of the total while RAYDAC, IBM 650 and others use self checking in varying lesser amounts.

---

1. J. Presper Eckert, "Checking Circuits and Diagnostic Routines", Convention Record of the IRE, 1953, 63.

On the other hand, programmed checks are used on almost all computers, even those with some built-in or self-checking circuits. The methods used to provide these programmed checks vary considerably since there is a great deal of flexibility in this approach. One obvious solution sometimes used is to do the problem on the computer twice and compare the answers. However, even if the operators may be persuaded to follow this procedure faithfully, it has some disadvantages and pitfalls. For instance, if the same error or errors are encountered each time the problem is run, the solutions, although incorrect, will agree. Even if the errors introduced are not repetitive, in a long computation much time may be wasted before the discrepancies are discovered. Another commonly used scheme is to prepare a special problem whose answers are known and to compare the machine results with the correct results. The comparison of these two sets of answers may be made either by the computer or by the operator. This sort of a problem is generally called a test routine and may be given to the machine before, after, or during an actual computation period. By specifically designing this routine so that it will detect the majority of possible machine errors a fair degree of assurance can be provided that the machine is working correctly.

Since the decision has previously been made to incorporate no self-checking circuitry in the WISC, the author will not attempt to present further arguments for or against either programmed checks or self-checking, but the interested reader is referred to several discussions of this topic.[2,3]

---

2. ibid., 62-65.

3. John W. Mauchly, "The Advantages of Built-in Checking", Proceedings of the Eastern Joint Computer Conference, 1953, 99-101.

2. Future Reliability

Some means of providing for reliable operation in the future must be employed. This can be accomplished through the use of preventive maintenance and the adoption of a tube and components program in an effort to locate and remove deteriorating components before they can cause a machine failure. Failures due to these gradually deteriorating components are called predictable faults as contrasted to sudden or unpredictable faults caused by shorts, opens, heater burn-outs, blown fuses, etc.,

One method of discovering the immuence of predictable faults in a computer is marginal checking. To illustrate the use of marginal checking as a preventive maintenance tool, let it be assumed that some circuit in the computer contains a vacuum tube whose plate resistance has increased appreciably from the design center value. The circuit containing this vacuum tube may still function correctly when the computer is running with normal heater or plate supply voltages. But, if either of these supply voltages is reduced sufficiently while the computer is operating on a self-checking program, the circuit can be caused to fail and the computer will very likely catch the error. The magnitude of the permissible supply voltage excursion is a measure of the margin of safety for this circuit, and the general practice of varying circuit operating conditions while checking for correct machine operation is called marginal checking.

Another method sometimes used in lieu of or in conjunction with marginal checking to discover deteriorated components involves the periodic removal and replacement of computer components and sub-assemblies for testing. Vacuum tubes, diodes, and other components can be tested either on conventional or special equipment to determine whether or not their

present characteristics fall within prescribed limits, while sub-assemblies can be tested by simulating actual operating conditions and checking for proper operation.

Additional benefits may be gained from the above procedures if all test and failure data is accurately recorded, since such data will provide a record of the behavior of all components and sub-assemblies in the computer. By analyzing this data the performance record of various crystal diode and vacuum tube types may be obtained and the products of different manufacturers compared. Such an analysis may also indicate points where improvement of circuit design or testing methods can increase the future reliability.

3. Restoring Reliability

Some procedure must be provided for quick restoration of the machine to reliable operation following the discovery that a fault exists. Some faults are very obvious and can be located almost immediately by human observation; however, in a machine as complex as a general purpose digital computer the location of a faulty component is often very difficult especially if the fault is of an intermittent nature. The approaches to this problem are of two types: those involving human observation of answers, waveforms, indicators, etc., and those involving the decision making ability of the computer to determine the location of its own fault.

The latter method makes use of test routines which not only ask the computer if it is giving correct answers, but seeks to determine the location of the faulty component from the answers received to different patterns of interrogation. Although the exact component at fault cannot generally be determined, the speed with which a fault may be localized

makes this method a great improvement over human observation. Obviously there are a number of difficulties involved in asking a computer that is operating incorrectly to correctly diagnose its own failures, but these to a large extent can be overcome by carefully designing the diagnostic routines with a complete understanding of the machine operation in mind.

B. Objectives

The work in connection with two of the tasks mentioned above; namely, debugging the WISC, and bringing it to operating condition, has been essentially completed. However, it still remains to provide some means of maintaining the WISC in operating condition a large percentage of the time. The author's goal in this thesis is to investigate some of the means by which this may be accomplished.

The first objective is to examine marginal checking as a preventive maintenance tool and to present some proposals concerning its use in the WISC. The second objective is to investigate the various test routines which will be required: Those which provide some indication of present reliability, those to be used in conjunction with marginal checking for preventive maintenance purposes, and those designed to aid in the location of faults so that the machine may be returned to reliable operation as quickly as possible.

C. Acknowledgments

The basic concepts--marginal checking and test routines for use with a digital computer--are not original with the author, and he wishes to acknowledge the ideas and suggestions received not only from the sources referred to in the text, but also from the many discussions--and sometimes lively arguments--with those with whom he has been associated with on this project.

He also wishes to thank all of the members of the staff of the Electrical Engineering Department for their encouragement, and especially Professor C. H. Davidson for his invaluable suggestions and assistance in the preparation of this thesis.

CHAPTER II

MARGINAL CHECKING

A. Purpose of Marginal Checking

In many electronic applications a certain number of intermittent errors of malfunctions can be tolerated due to the redundancy of the information being handled; however in an electronic digital computer a single error may spread through all subsequent calculations in the program. Since a computer such as the WISC may process about 2×10^4 orders per hour with each order involving 10^6 operations such as flip-flop reversals and gate samplings, it can be estimated that each hour of error free computing time requires 10^{10} such successful operations. The magnitude of the reliability problem can easily be estimated from an inspection of two factors: average component life, and the number of such components. For instance, assume a system containing 10,000 vacuum tubes and 5,000 crystal diodes with average life expectancies of 5,000 hours each. If these components are not replaced until they fail, simple arithmetic will show an expected average of three failures per hour. In the WISC there are presently about 1,500 vacuum tubes and 350 crystal diodes. Assuming life expectancies of 5,000 hours each, a predicted average of three failures per eight hour computer day would not allow satisfactory computation of large scale problems.

As mentioned in the introduction to this thesis, components may be removed and tested periodically in order to discover and replace those which have deteriorated before they can cause trouble; however, there is a limit to the frequency with which this can profitably be done. To test all of the tubes in the WISC each 10 hours (for example) would require that 1,500 vacuum tubes be tested during the 10 hours of

computing time. Not only would the amount of time spent testing components be excessively large, but also the average life expectancy of the tubes would be decreased by all of the handling necessary. Nevertheless, periodic testing, although not as frequently as every 10 hours, was used to keep the computer running during the testing and debugging stage when complete programs could not be run on the machine.

Marginal checking can provide the same check on some of the components of the system without requiring that they be removed. Its chief purpose is to detect the existence of components whose characteristics have changed to such an extent as to be suspected of imminent failure, the assumption being that any further change will be in the direction of a failure. In addition to detecting a future failure, marginal checking can assist in locating the marginal component, since operating conditions of small sections of the computer can be varied in turn while normal operating conditions are maintained in the rest of the computer.

B. Design Considerations

In relying on marginal checking, two assumptions are made. First, a large number of machine faults are due to steadily deteriorating components and therefore are predictable. According to the published experience with other computers[1,2] this would seem to be true. Second, it is assumed that the marginal checking scheme which is to be used adequately tests the components of the computer. Obviously varying one operating condition may not test all circuits in a section of the computer;

---

1. Maurice V. Wilkes, Montgomery Philster Junr., and Sidney A. Barton "Experience With Marginal Checking and Automatic Routining of the EDSAC" Convention Record of the IRE, 1953, 71.

2. Norman H. Taylor "Marginal Checking as an Aid to Computer Reliability" Proceedings of the IRE, 38-12 (1950), 1421.

some circuits may be much more sensitive than others to changes in this particular variable, while others are affected very little. For example, a flip-flop may be made to fail by reducing the plate supply voltage by say 35 percent, while a pentode amplifier in the same computer section may not fail until the plate supply voltage is reduced by 50 percent. Since the plate supply voltage cannot be used to test the condition of the pentode, an additional variable, perhaps the screen voltage supply, must be used.

Ideally marginal checking should be incorporated in the original design of a computer. If marginal checking is applied to the proposed basic circuits when they are in the breadboard stage, not only can the most effective means of testing the components be ascertained, but in addition some circuit shortcomings may be uncovered which might not otherwise come to light until much later. Furthermore, with the variables decided upon, a system for providing the changes in these variables can be built into the machine, and at this stage a good logical subdivision, with respect to marginal checking, can be made. On the other hand, when marginal checking is added to an existing machine the mechanism for varying the parameters must be compatible with the already existing circuits, and the subdivision of marginal checking lines will be dictated considerably by power supply arrangements and mechanical or physical factors. A discussion of the philosophy followed in incorporating marginal checking into the original design of a computer and in using marginal checking as an aid in circuit design can be found in the literature[3].

---

3. Raymond E. Nienburg, "Reliability of an Air Defense Computing System: Circuit Design, "IRE Transactions on Electronic Computers, EC-5 (1956) 227-233.

No provisions for marginal checking were included in the original design of the WISC. However, during a stage of its development, when the d.c. power supply system had to be moved and enlarged and it was decided to use well regulated generators to provide the main d.c. power for the WISC, the author undertook to redesign the power supply incorporating provisions for independently varying each d.c. power distribution line. These marginal checking facilities proved to be of considerable help in testing and debugging the computer, and some minor changes in basic circuitry were found to be desirable as a result of their use.

C. Effect of Variables

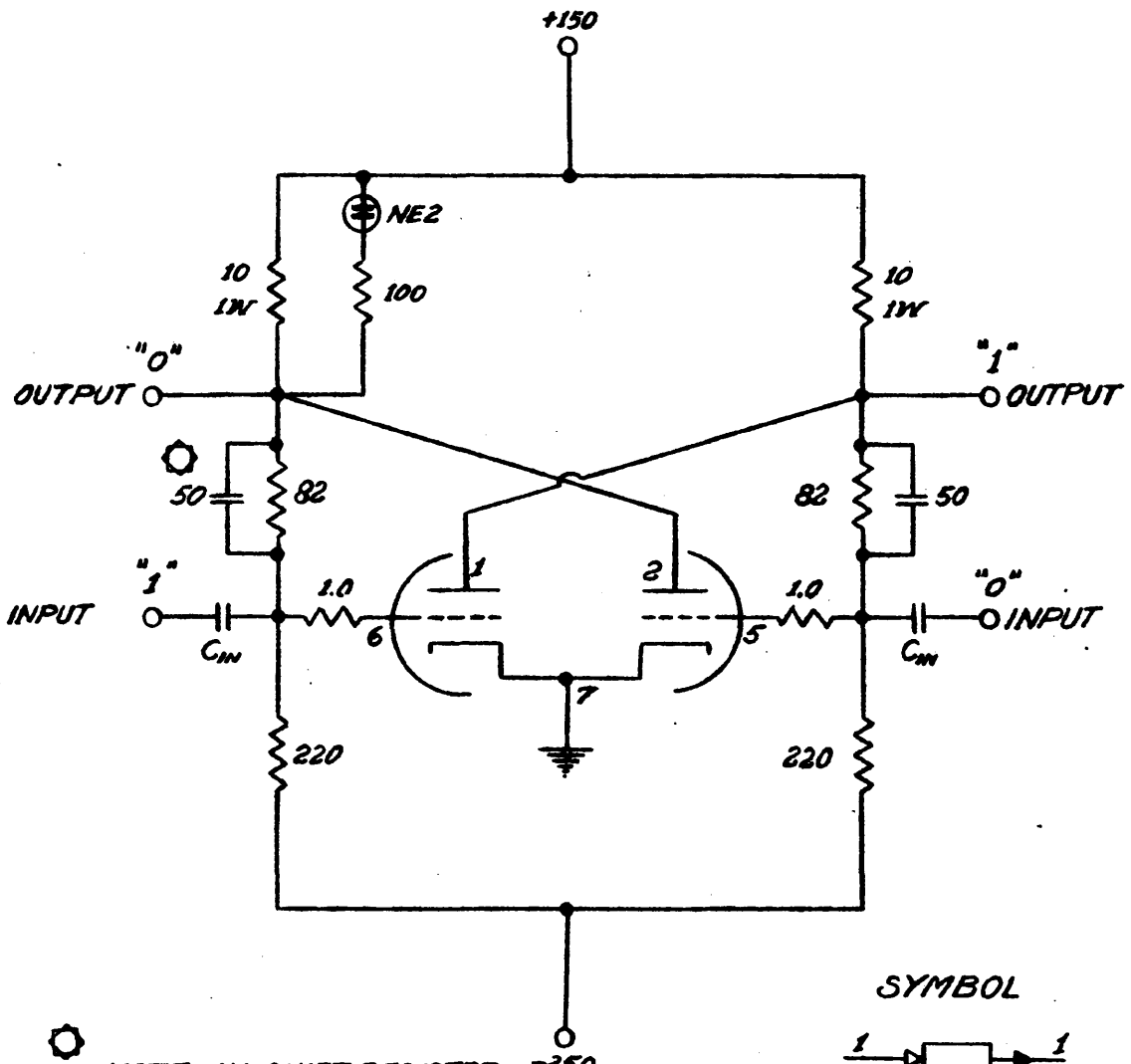
Of the many possible variables such as heater voltages, d.c. power supply voltages, signal bias, signal magnitude, etc., it seems feasible that in the WISC it will suffice to allow for variation of heater and d.c. power supply voltages only. A prime consideration in the selection of the variables to be used for marginal checking is the effect of each variable on the circuits to be used. Consider first the effect of the d.c. supply voltages on the basic WISC circuits.

1. D.C. Supplies

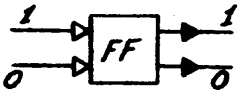
a. Flip-Flop

It can be seen from an examination of Figure 1 that the steady-state voltages at all of the tube terminals depend on the characteristics of the tube. For the sake of discussion assume that the tube is in the "one" state, with the triode on the right conducting. The voltage at the plate of the left hand triode is least affected by the tube conditions and is nominally 133 volts, while the voltage at the conducting grid varies with the tube type and characteristics from +0.5

Figure 1



SYMBOL



NOTE: IN SHIFT REGISTER FLIP-FLOPS USE 30 MM

C_{IN} = 30 MM IF INPUT IS PULSES

C_{IN} = 10 MM IF INPUT IS FF

CAPACITORS IN MMfd
RESISTORS IN KΩ ½ W, 10%
} UNLESS OTHERWISE NOTED

STANDARD WISC FLIP-FLOP
TYPE 6J6 or 5844
DRAWING No IV-021S,L-03A
ISSUE NUMBER 2
DATE: 7-27-57

to +1.5 volts. The other two electrode voltages vary considerably from tube to tube; for example, the voltage at the conducting plate may vary from +40 to +75 volts with corresponding non-conducting grid voltages of -38 and -13 volts respectively.

Now assume that for the above conditions the B-(-250 volt) supply were changed by 10 percent to -225 volts. This would make the voltage at the non-conducting grid -6 volts for the case if +75 volts on the conducting plate, and the circuit probably would not operate correctly. Obviously a tube which would give a conducting plate voltage as high as 75 volts would not deliberately be placed in this circuit (see Chapter IV), but if the emission of some tube in use had deteriorated to this condition, varying the B- by the 10 percent mentioned would cause it to fail thus giving an indication of its deteriorated condition. Other defects such as triodes exhibiting remote cut-off characteristics, and high resistance grid-to-cathode shorts which would allow marginal operation in this circuit under normal conditions, would also cause failure if the B- supply were used as a marginal checking variable.

Again examine the flip-flop circuit, this time considering the effect on the circuit of varying the B+(+150 volt) supply. Assuming that this supply voltage is increased from +150 to +160 volts, the voltage at the non-conducting (off) plate would rise from 133 to 142 volts while the voltage at the conducting grid would increase only slightly due to the clamping action of the grid. However, this increase in grid voltage tends to decrease the plate voltage of the conducting tube while the increase in the B+ tends to increase it. Figure 2 shows a plot of the average conducting plate voltage versus B+ for

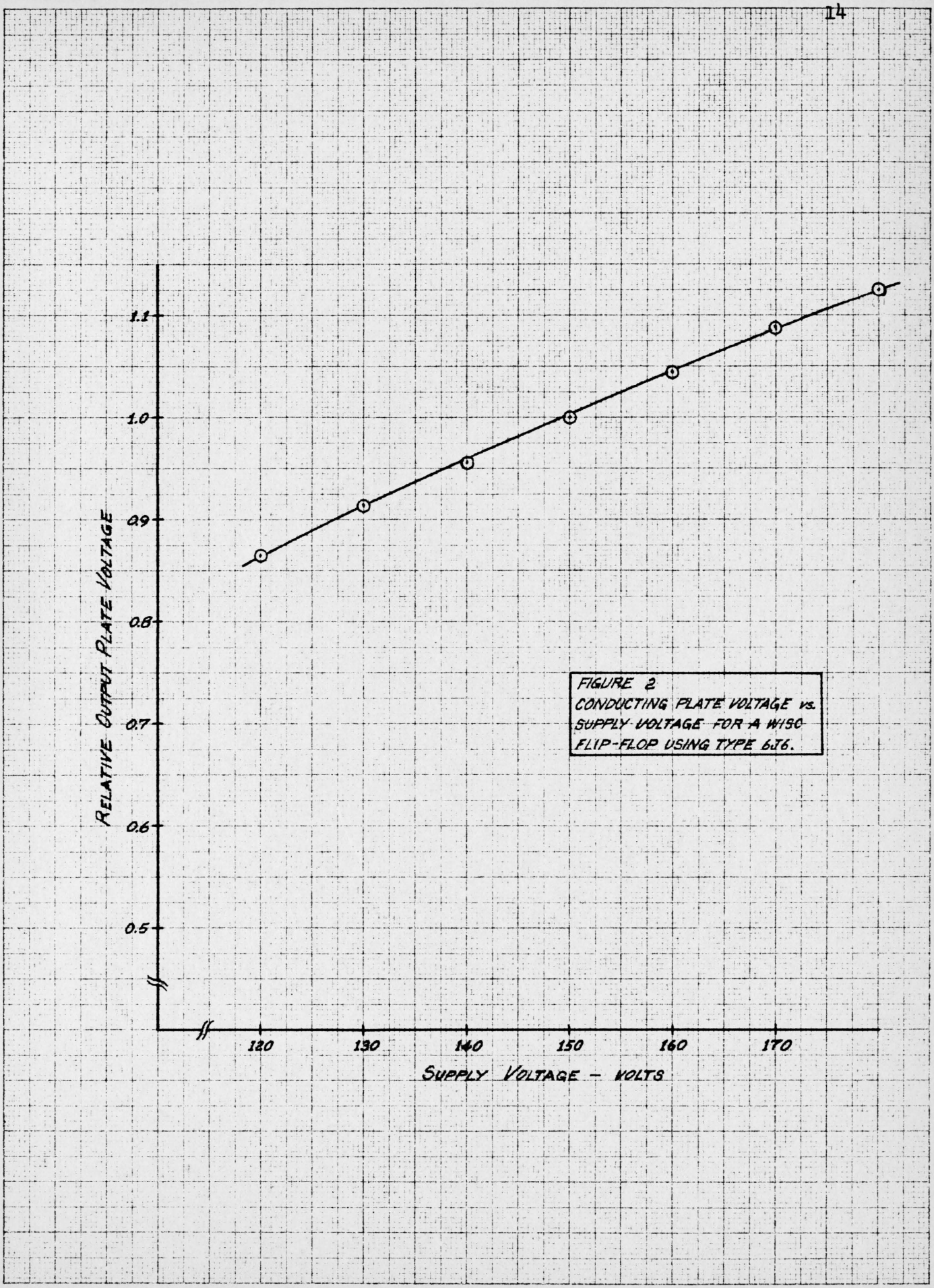


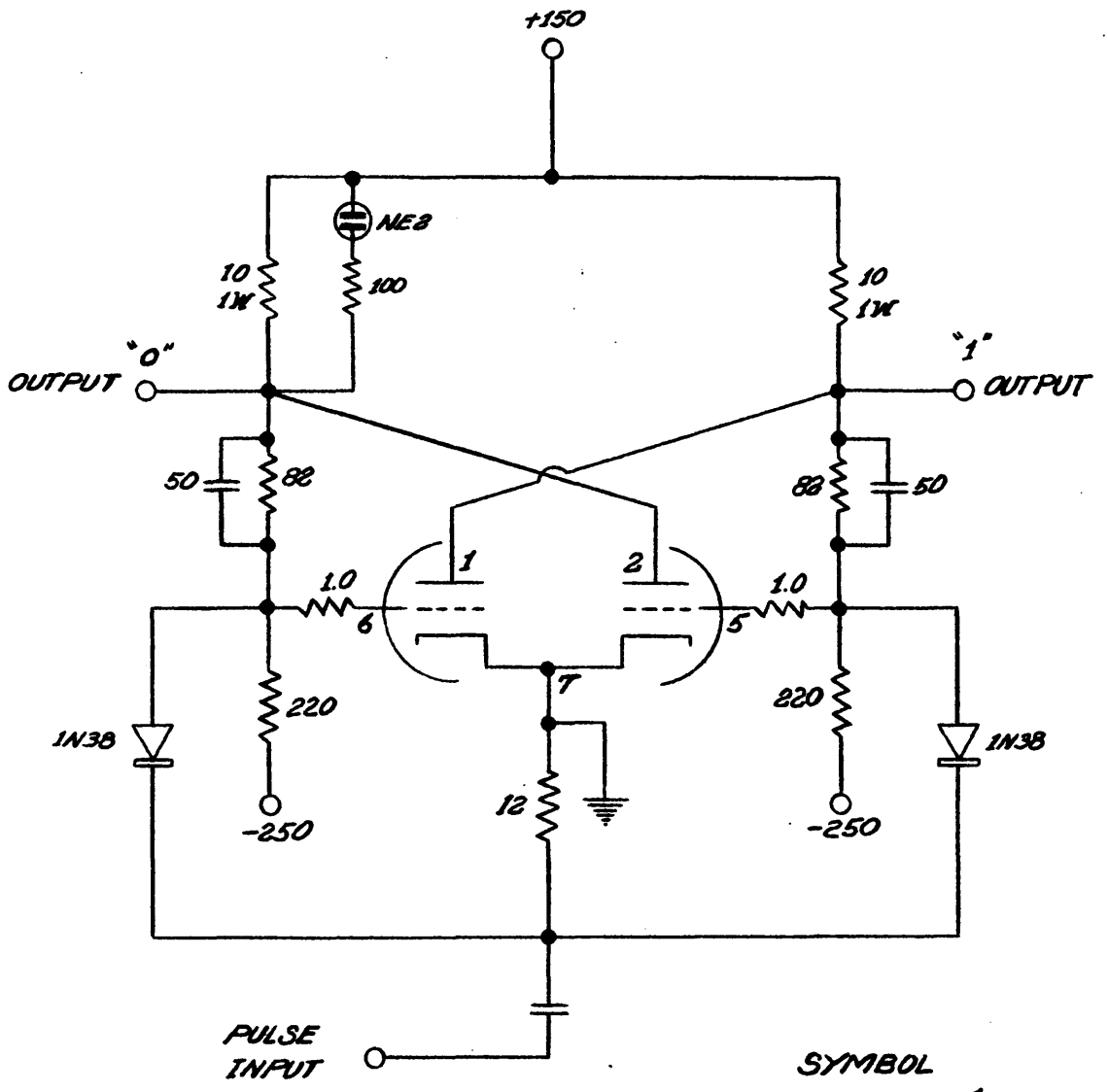
FIGURE 2
CONDUCTING PLATE VOLTAGE vs. SUPPLY VOLTAGE FOR A WISC FLIP-FLOP USING TYPE 6J6.

several 6J6 double triodes in the flip-flop circuit. From an examination of this plot it can be seen that the voltage at this plate is not greatly affected by B+ variation and that a relatively large change in B+ is required in order to make the flip-flop fail. Due to the design of the pulsed gates used in the WISC (see part c-Gates) large changes in the voltage at the non-conducting plate will cause the gate circuits to fail. Therefore, the +150 volt supply cannot be used as a marginal checking variable to test the vacuum tubes in flip-flop circuits. For these reasons the marginal checking variable to be used to check flip-flop circuits will be the -250 volt supply.

b. Counter Circuits

An examination of Figure 3 will show that the flip-flop and counter circuits are identical except in the method of triggering. The steady-state voltages in these two circuits would be identical if it were not for the reverse leakage current of the non-ideal diodes used for triggering the counter circuit. These diodes are tested before installation and will have measured leakage currents up to 100 microamps with 75 volts applied. Since the reverse voltage in these circuits is nominally about 30 volts the leakage current will be even less, and the presence of these diodes should not cause any significant difference between the steady-state voltages in the counter and flip-flop circuits. However, as the diodes deteriorate the reverse current increases tending to make the negative grid in the circuit more positive which is a change in the direction of circuit failure. If the reverse current of some diode in use in a counter circuit increases to the point where

Figure 3



CAPACITORS IN MMfd
RESISTORS IN KΩ ½ W, 10% } UNLESS OTHERWISE NOTED

STANDARD WISC COUNTER
TYPE 6J6 OR 5844
DRAWING No W-022S, L-01A
ISSUE NUMBER-2
DATE: 7-27-57

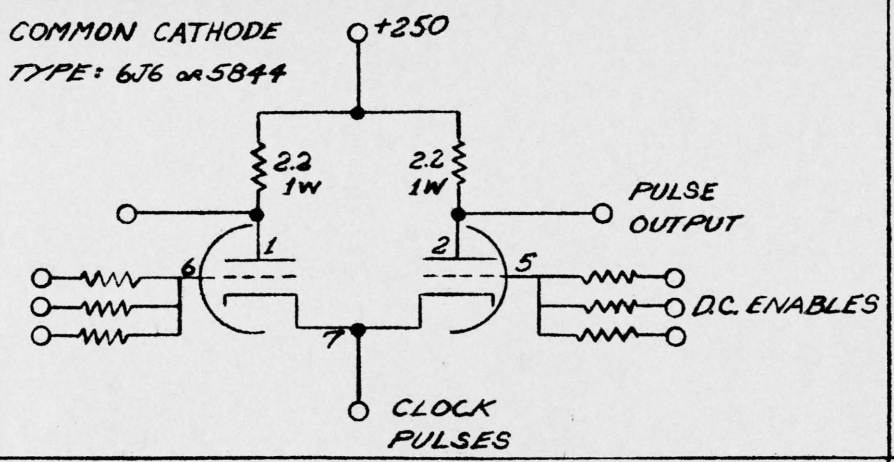
the circuit is sufficiently close to failure under normal conditions, making the -250 volt supply more positive will cause the circuit to fail, indicating a defective component. The failure of a counter circuit during marginal checking may indicate the deterioration of a diode, vacuum tube, or a combination of these two.

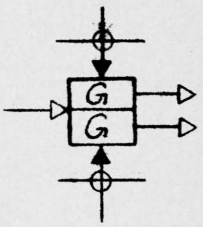
c. Gate Circuits

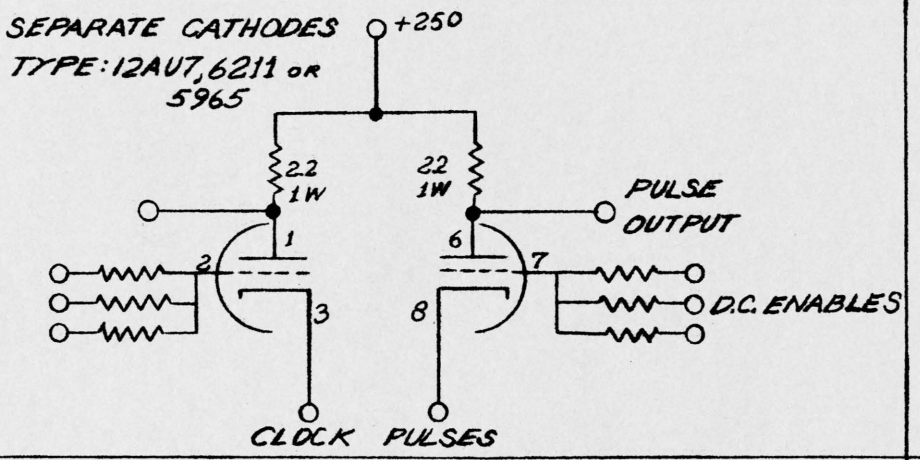
Almost all of the gate circuits in the WISC are of the pulsed gate type shown in Figure 4. The three "enables" or inputs to the grid are said to be flip-flop derived; that is, they are directly connected either from flip-flop plates or from circuits which approximate the same steady-state levels as a flip-flop.

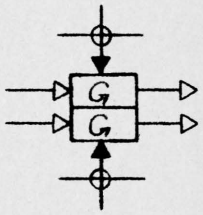
Assume that the three inputs to a gate are connected to the one's output of flip-flops A, B, and C which may be at either +130 or +55 volts depending upon the state of the flip-flops. Then the voltage at the control grid of the gate tube will be +130 volts if the one's output of all three flip-flops is high, 105 volts if two are high, 80 volts for one high, and 55 volts for the case of none high. The signal connected to the cathode of the gate is a 1 microsecond wide negative pulse biased at +150 volts, the repetition rate depending upon the application. The magnitude of this negative pulse, about 30 volts, drives the cathode down to approximately 120 volts, and if the control grid is at the 130 volt level the tube will conduct, causing a one microsecond pulse of about 30 volts magnitude to appear at the plate of the gate. If, however, the control grid is at any of the lower levels during the time that the cathode is pulsed the

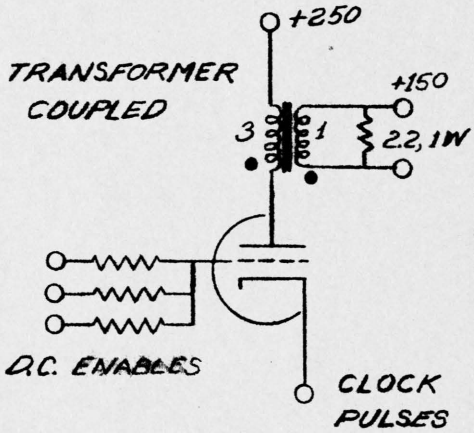
Figure 4



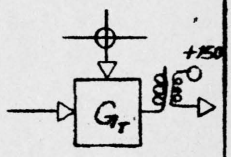








NOTE THIS GATE IS USED FOR BOTH POSITIVE AND NEGATIVE OUTPUT, WITH SEVERAL BIASES ON THE SECONDARY, AND WITH PULSE XFMRS OTHER THAN THE 3:1 RATIO SHOWN.



RESISTORS IN KΩ, ½ W, 10%
ALL GRID RESISTORS ARE 100 K, ½ W

PULSED GATES - WISC
DRAWING Nº W-023S, L-04A
ISSUE Nº 2
DATE: 7-31-57

control grid will be biased well beyond cut off, 15 volts or more, and the tube will not conduct.

When discussing the flip-flop circuit and the use of the 150 volt supply as a marginal checking variable the comment was made that large changes in the voltage at the non-conducting plate will cause the gate circuits to fail; this can be readily shown with an example. Assuming that the +150 volt supply for flip-flop A, B, and C were increased to say 170 volts, the four voltage levels possible at the gate grid would be approximately 150, 119, 88 and 57 volts, and the gate would not operate correctly at two of these voltage levels if the cathode pulse bias remained at +150 volts. If the grid were at the upper level, +150 volts, the tube would conduct continuously, possibly exceeding the plate dissipation of the tube or the dissipation rating of the plate resistor, and greatly reducing the output signal when the cathode is pulsed. The gate would also emit pulses not only when the cathode is pulsed with the grid at 119 volts, but also even in the absence of cathode pulses whenever the grid level changed from 119 volts to 150 volts. For a negative change in the B+ to the flip-flops, the effect is still to make the gates fail. If the B+ were reduced from +150 to +130 volts the highest gate grid level would be 116 volts, and since the gate cathode is only pulsed down to +120 volts, the magnitude of the signal at the plate would be decreased greatly.

With the preceding discussion in mind it can be seen that varying either the +150 volt supply to the flip-flops connected to a gate, or changing the bias of the cathode pulses, can cause a great deal of change in the magnitude of the signal occuring at

the plate of the gate tube. It would seem, then, that either of these two voltages might be used as a marginal checking variable; however, a closer examination must be made of the effects to be expected when varying these voltages. Since an excursion of the plate supply voltage to the flip-flops which are sampled by the gate and an excursion of the pulse bias in the opposite direction produce approximately the same results, only one of these variables, the pulse bias, will be considered.

Assuming that the pulse bias is gradually made more positive, what changes can be expected in the operation of the circuit during the time the cathode is pulsed? First, since the tube normally operates in the positive-grid region during this time, a decrease in grid current will occur tending to reduce the amplitude of the output pulse. However, this reduction will not be very great since the grid current, approximately 1/3 milliamperes under normal conditions, is so small as to have relatively little effect upon the magnitude of the plate current. Second, with further increase in pulse bias, circuit operation will pass into the negative-grid region where a large decrease in plate current will occur. Since all gates will show a marked decrease in output pulse size as this negative-grid region is entered, this region of pulse bias variation has no significance for the purposes of marginal checking.

A variation of pulse bias in the negative direction will first increase the output pulse magnitude by increasing the grid current, then decrease the output pulse magnitude by allowing the tube to conduct when the cathode is not pulsed. In addition pulses will appear at the gate output for grid levels other than the highest or 130 volt level.

In summary, the only useful range of this variable is between the points where the grid is biased negatively and where the tube begins to conduct continuously; furthermore the output pulse magnitude is changed very little as the variable goes between these limits. Then,since the main cause of computer failure due to deterioration of components in this circuit is insufficient pulse magnitude brought on by a reduction of tube current, cathode bias is not a good variable to use for marginal checking purposes.

The next question to be asked, then, is: What will be the effect on the operation of the gate if the B+ supply to the gate is varied from its nominal value of +275 volts? The answer can be seen from an examination of the plot of the load line on the plate characteristics of the tube. Because the signal is cathode coupled, the effective plate supply voltage changes from +125 volts to +155 volts when the cathode is pulsed. Since the grid current is small the output signal swing may be approximated by assuming the grid swing is from below cut-off to zero-bias. Using these figures and the plate characteristics of a 12AU7 triode, one of the tubes presently used for gate applications, the output pulse magnitude is found to be 33 volts. Using the same data, it is found that if the B+ is decreased by 30 volts, the output pulse magnitude will decrease from 33 volts to 25 volts, a 25 percent change, with the change between these two values being almost linear. Since the effect of such a change simulates the reduction in current expected from a deterioration of the tube, and since the control afforded is sufficient and gradual, this marginal checking variable has been selected to be used for this circuit.

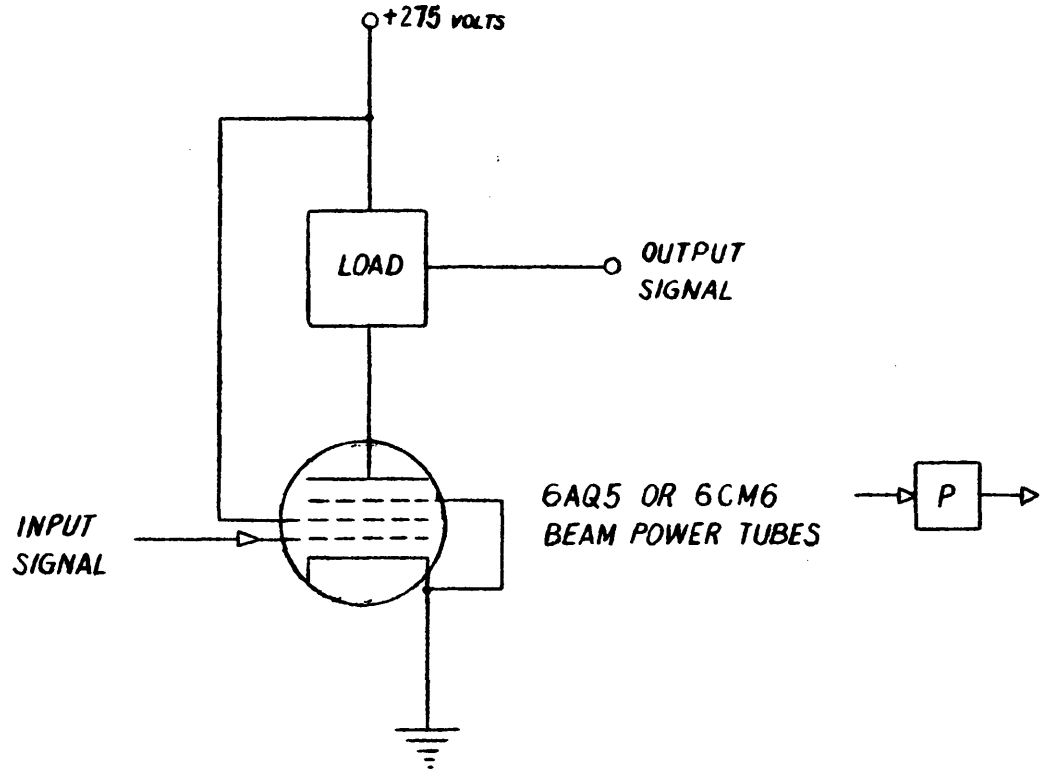
d. Pulse Amplifiers

The pulse amplifier circuits used in the WISC are mostly of the type shown in Figure 5, although a number of the transformer coupled gates of the type shown in Figure 4 exist. Since the discussion for the normal type of gate can be extended to cover the transformer coupled gate, only the pulse amplifiers similar to Figure 5 need be considered here.

The load shown as a box in the figure may be one of several pulse transformers, a delay line, or the write winding of a magnetic drum head, depending upon the application. This discussion would become needlessly complicated if the transient phenomenon associated with each of these loads were discussed; therefore, only ideal transformers and resistive load lines will be assumed. The signal at the grid is a positive going pulse obtained through a pulse transformer biased at -60 volts, and with the beam power tube not in the circuit the positive-going portion of this signal will be approximately 100 volts in magnitude. Thus when the circuit operates normally the grid will be driven from -60 volts to a positive voltage dependent upon the tube characteristic and the impedance of the source of the input signal.

The loads are of two types: Those which give an operating line which ends on the plot of the plate characteristics of the tube below the knee of the curve, and those which do not. In the first case the plate is "bottomed". This tends to stabilize the amplitude of the plate swing and hence the output pulse, so that tube deterioration, within limits, has relatively little effect on the output pulse magnitude. Assuming that deterioration of the beam power tube can be adequately

Figure 5



PULSE AMPLIFIER - WISC
DRAWING NO. W-027S-02A
ISSUE NO. 2
DATE: AUG 12-57

represented by a compression of the plate characteristics toward the plate voltage axis, it can be seen that the output pulse will be fairly constant up to a point, then fall rapidly with further deterioration of the tube.

Compression of the plate characteristics toward the plate voltage axis will also result from a decrease in the applied screen voltage; consequently a decrease in screen voltage can be used to simulate further deterioration of the tube. Figure 5 shows screen and plate load connected to the same B+ line. Now if the B+ is decreased, the output pulse amplitude may decrease for two reasons: One, a reduction in plate supply voltage to the tube reduces the available plate swing and therefore the output pulse size; two, as mentioned above, the reduction of screen voltage may reduce the output pulse size. Since all of the pulse amplifier circuits now in use have been constructed with the same B+ line connected to the load as is connected to the screen, this variable is being used at the present. Once adequate data has been gathered on the effectiveness of varying both the plate supply and screen voltages together, a decision will be made as to whether or not these circuits should be modified to allow independent variation of each.

The above discussion can be extended to cover the case where the lower impedance loads do not cause a bottoming of the plate in which case the output pulse size will decrease directly either as the screen voltage is reduced or as the tube deteriorates.

Before settling upon the B+ supplies as the marginal checking variable for use with pulse amplifiers, it would be advisable to examine the other available variable, the -60 volt bias to the

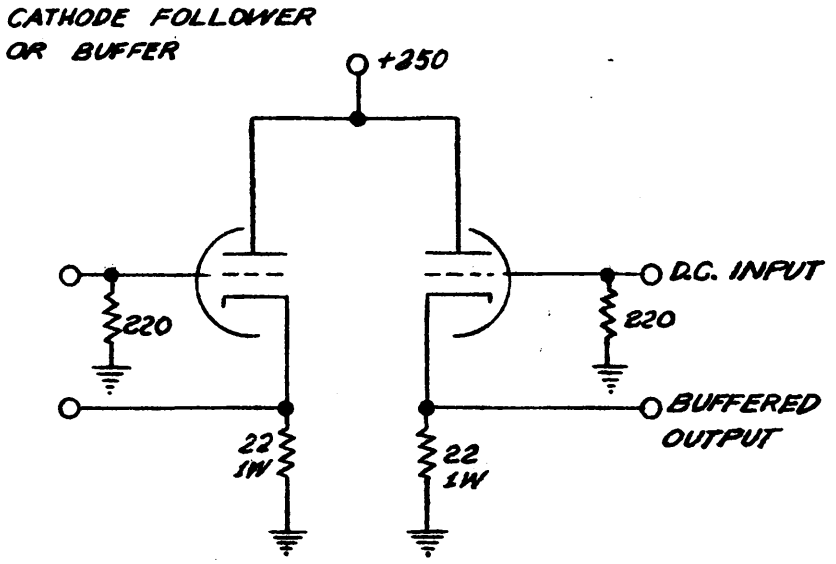
pulse transformers. It has been stated that the waveform at the grid is a positive going pulse of approximately 100 volts obtained through a pulse transformer biased at -60 volts. As is characteristic of transformer coupled waveforms, the signal is not square but exhibits a finite rise and fall time, and in some instances superimposed shock oscillations and droop. Consequently, it would not be advisable to increase the bias on these transformers to simulate tube deterioration since the reduction in output pulse magnitude would be accompanied by a change in the pulse width and wave shape.

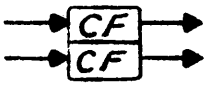
e. Other Circuits

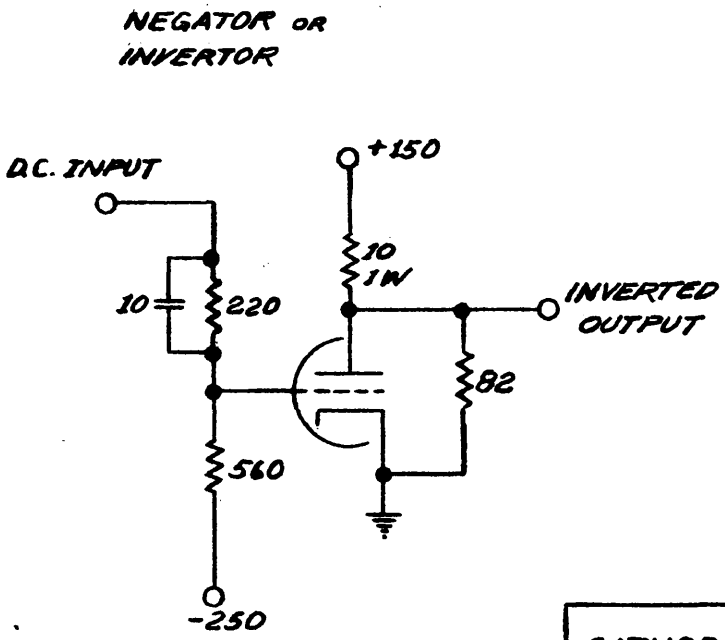
Figure 6 shows both the standard cathode follower and the negator circuits used in the WISC. In most circuits a deterioration of the vacuum tube is thought of as producing a decrease in cathode current; however, in the cathode follower circuit the current, for a given input level, remains relatively constant so that deterioration of the cathode causes instead a decrease in tube bias. As the cathode emission decreases, a point will be reached where the cathode follower draws sufficient grid current to cause the source of the input signal to fail. Since a similar effect can be brought about by a reduction in the plate to cathode voltage, the B+ (275 volt) supply has been selected as a marginal checking variable for this circuit.

The negator circuit used to invert or negate a flip-flop signal gives an output which is at one of the two nominal voltage levels, +133 or +55 volts. As in the case of the flip-flop the upper level is fixed while the lower level is subject to the characteristics of the vacuum tube. A reduction in cathode current sufficient to cause

Figure 6







CAPACITORS in MMfd. } UNLESS
RESISTORS in KΩ, ½ W, 10% } NOTED

CATHODE FOLLOWER &
NEGATOR - WISC
DRAWING No W-024S, L-02A
ISSUE No 2
DATE: 7-31-59

a flip-flop to fail will not cause a failure of the negator circuit itself, although it may cause a failure in the gates using the inverted signal. Since large variations in the +150 volt supply cannot be tolerated because of the gate circuits, and since variation in the -250 volt supply will not give the desired effect, these power supply variables cannot be used for marginal checking of this circuit. Consequently variation of heater voltage which will be discussed next must be relied upon to check the negator circuit.

As the remainder of the circuits used in the WISC are few in number and are either of special types or variations of those already mentioned, these special circuits will be checked using the variables discussed above.

2. Heater Voltage

Whereas in the preceding section the effect of the d.c. variables on both the circuit and the components was discussed, the effect of a reduction in heater voltage on only one component, the vacuum tube, will now be considered. Since the majority of tube failures in computers can be attributed to a deterioration of the emission characteristics[4], the main concern here will be the heater sensitivity -- the reduction of positive-bias plate current due to a decrease in heater voltage -- of the double triodes being used or considered for use in the WISC.

---

4. J. A. Goetz and H. J. Geisler, "Electron Tube and Crystal Diode Experience in Computing Equipment", Proceedings of the Eastern Joint Computer Conference, 1953, 61-72.

The following figures show plots of the emission characteristics of groups of 6J6 tubes as a function of heater voltage. The ordinate I_b, relative plate current, is defined by the expression

$$I_b = \frac{(I_b)_{E_h}}{(I_b)_{6.3}}$$

where $(I_b)_{6.3}$ is the positive bias plate current with 6.3 volts applied to the heater, and $(I_b)_{E_h}$ that current for a heater voltage E_h. The measurements were taken in a circuit which duplicates the conditions found in a WISC flip-flop; that is, the grid current is 0.45 ma while the equivalent plate load resistor and B+ supply are 8.9 kilohms and +133 volts respectively.

Figure 7 shows a plot of I_b vs E_h for 10 tubes, 20 cathodes, that have been aged for 200 hours in aging equipment specially designed for use with the tube program. The solid line represents the average of all 20 cathodes while the dotted lines represent the maximum deviation from the average. It can be seen that the plate current for the average tube decreased by 4.7 percent when the heater voltage was reduced by 32 percent to 4.3 volts. This is a small change in I_b since it would cause the conducting plate voltage of a flip-flop to rise from say +55 volts to only +58.7 volts. On the basis of this data, decreasing the heater voltage to simulate tube deterioration does not appear very practical since a large change in the variable produces such small results.

However, before abandoning this as a marginal checking variable, it would be advisable to examine groups of tubes that have seen more use. Although the WISC tube program has unfortunately been in

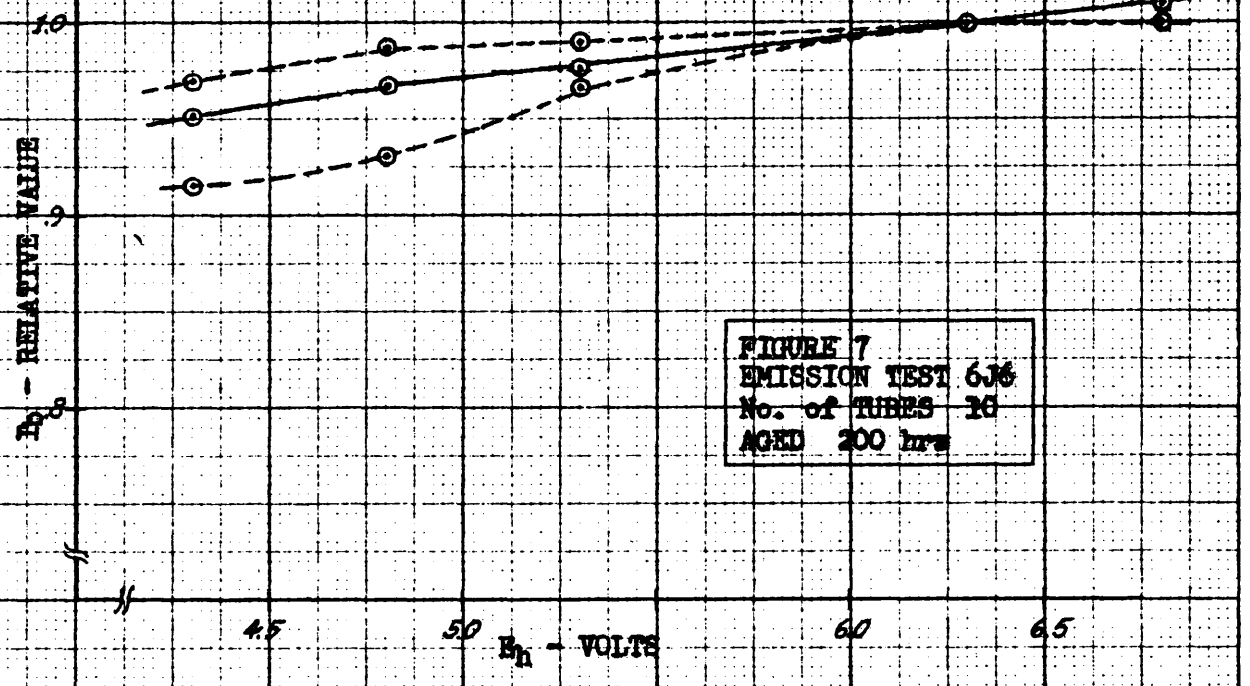


FIGURE 7
EMISSION TEST 6J6
No. of TUBES 20
AGED 200 hrs

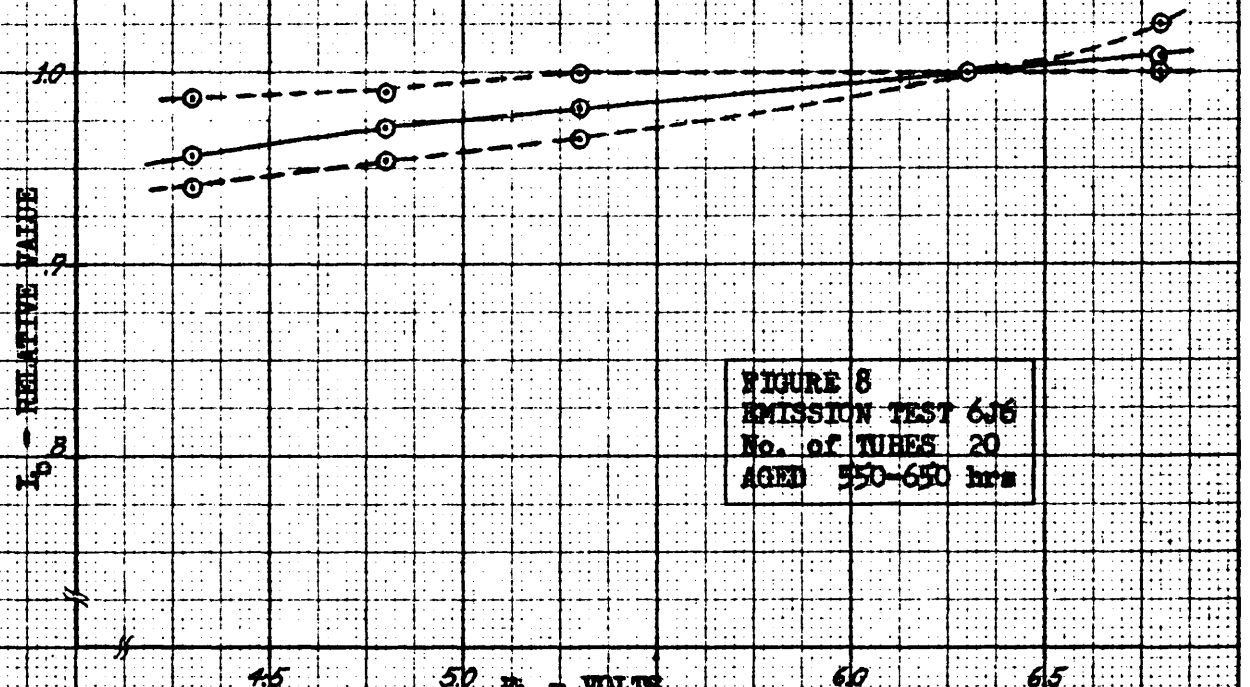


FIGURE 8
EMISSION TEST 6J6
No. of TUBES 20
AGED 550-650 hrs

existence for a relatively short time, making it impossible to determine the time in service for many of the older tubes in use, it was possible from the tube program data to locate a group of tubes that has been in use in the computer for 550 to 650 hours and to obtain sufficient data from them to plot Figure 8. For this group of twenty tubes the same 32 percent reduction in heater voltage decreased the plate current of the average tube by only 4.2 percent; consequently, the same conclusion is reached as before; namely, that plate current is relatively unaffected by reasonable variations in heater voltage. This should not be surprising since this is one feature of vacuum tubes which makes them reliable and consequently has been demanded of tube manufacturers.

However, one thing has not yet been settled; i.e., what happens to these characteristics as the tube approaches the end of its useful life? As mentioned previously, sufficient information about time in service is not presently available; however, data has been collected for a large number of the 6J6 tubes in the WISC that have developed cathode poisoning. These tubes which were purchased as war surplus material have well over 1000 hours of service since they were part of the first group of tubes to be used in the WISC. Although they are all manufactured by the same firm, this cannot be described as data from a controlled experiment since time in use, shelf life, date of manufacture, and degree of cathode poisoning all vary greatly from tube to tube. Nevertheless, the data is of sufficient interest that it has been plotted in Figure 9.

Two things can be seen from an examination of this figure: First, the plate current of the average tube in Figure 9 drops off

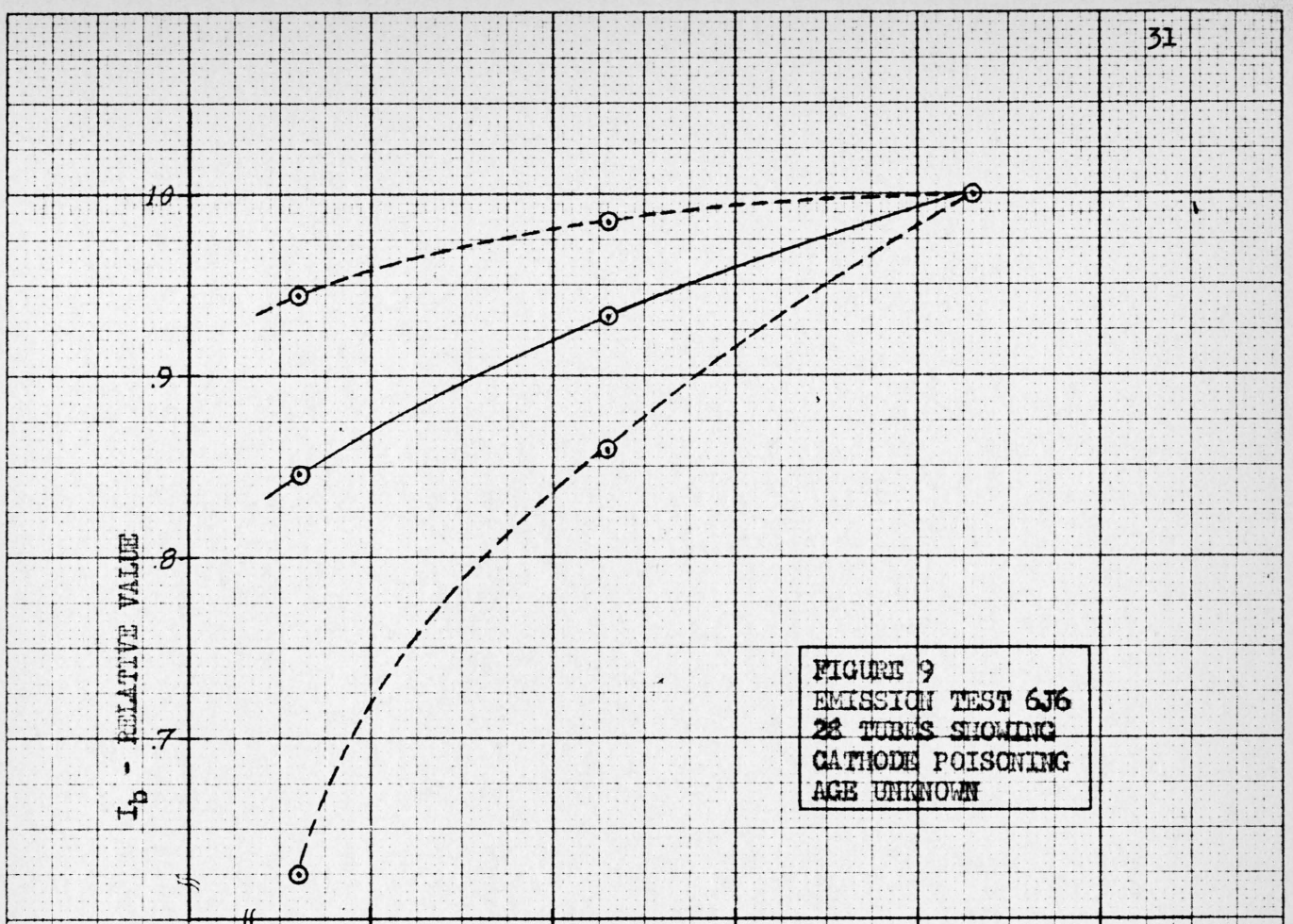


FIGURE 9
EMISSION TEST 6J6
28 TUBES SHOWING
CATHODE POISONING
AGE UNKNOWN

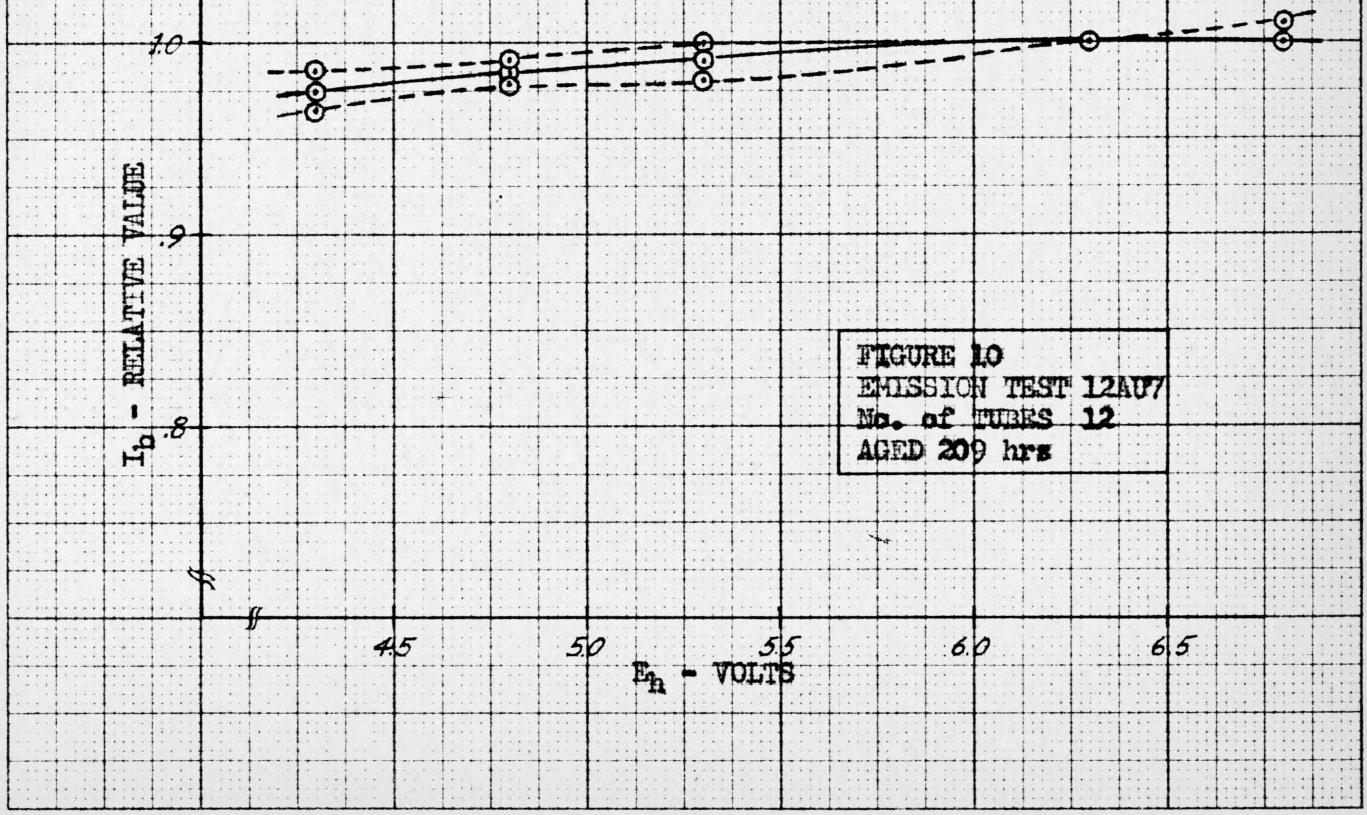


FIGURE 10
EMISSION TEST 12AU7
No. of TUBES 12
AGED 209 hrs

rapidly with a decrease in heater voltage. Second, there is a large difference between the maximum and minimum plate current values for tubes in this group. The second observation may have little significance since, as has already been pointed out, the life histories of these tubes differ considerably, but the first would seem to indicate that a slump or fall off in plate current at reduced heater voltages does develop as the tubes approach the end of their useful life. Even though these tubes show a much greater heater sensitivity, the average change in I_b for this case is still only 15 percent for a 32 percent change in E_h, while for a more reasonable decrease in E_h, say from 6.3 to 5.3 volts, the average change for these tubes would be only 7 percent. Consequently, as a marginal checking variable the heater voltage would not seem to be as effective as the d.c. supply voltages previously selected.

Naturally, other tube types than the 6J6 should be examined before any concrete conclusions are reached. For this reason data on several other tube types has been gathered and is presented in Figures 10, 11, and 12. The curves for type 12AU7, Figure 10, and the curves for type 5965, Figure 11, seem to indicate that both of these types are even less sensitive to variations in heater voltage than the 6J6 while the curve for the 6BQ7A, Figure 12, shows a heater sensitivity comparable to the 6J6. Figure 12 also shows the difference in heater sensitivity between a group of tubes that have seen 200 hours of use and a group, presently being life tested under normal operating conditions, having 5241 hours of use, and lends confirmation to the suspicion that heater sensitivity increases slowly with age.

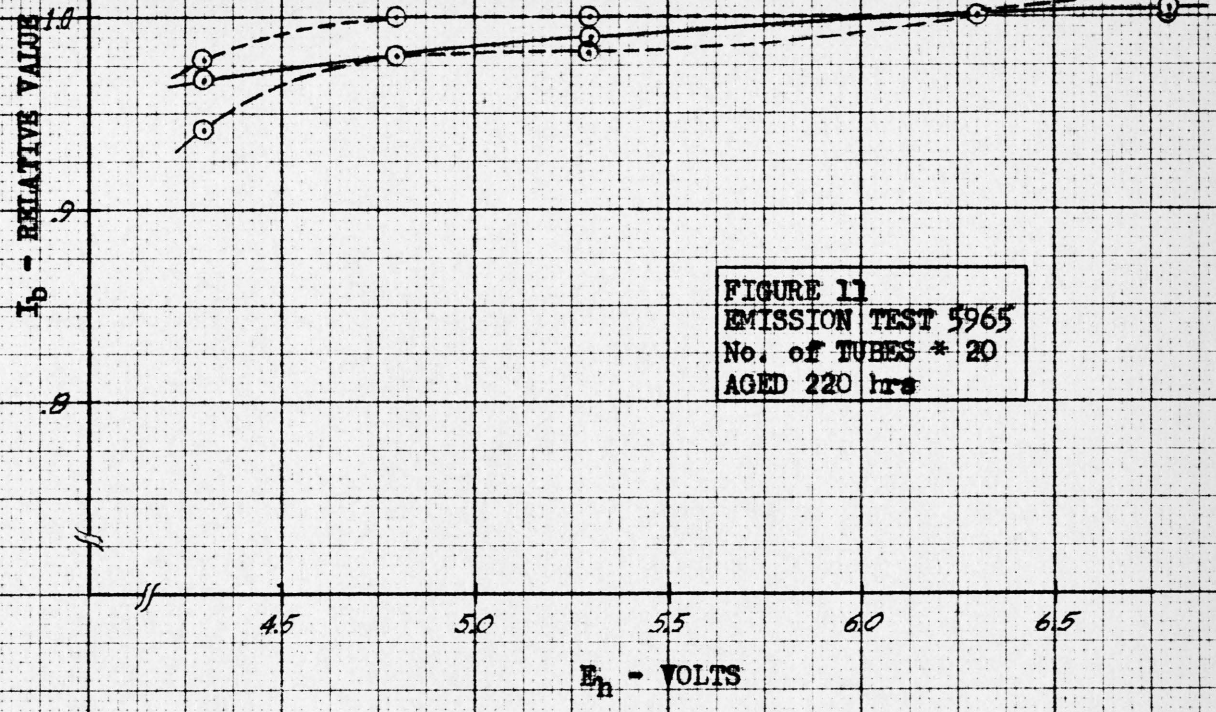


FIGURE 11
EMISSION TEST 5965
No. of TUBES * 20
AGED 220 hrs

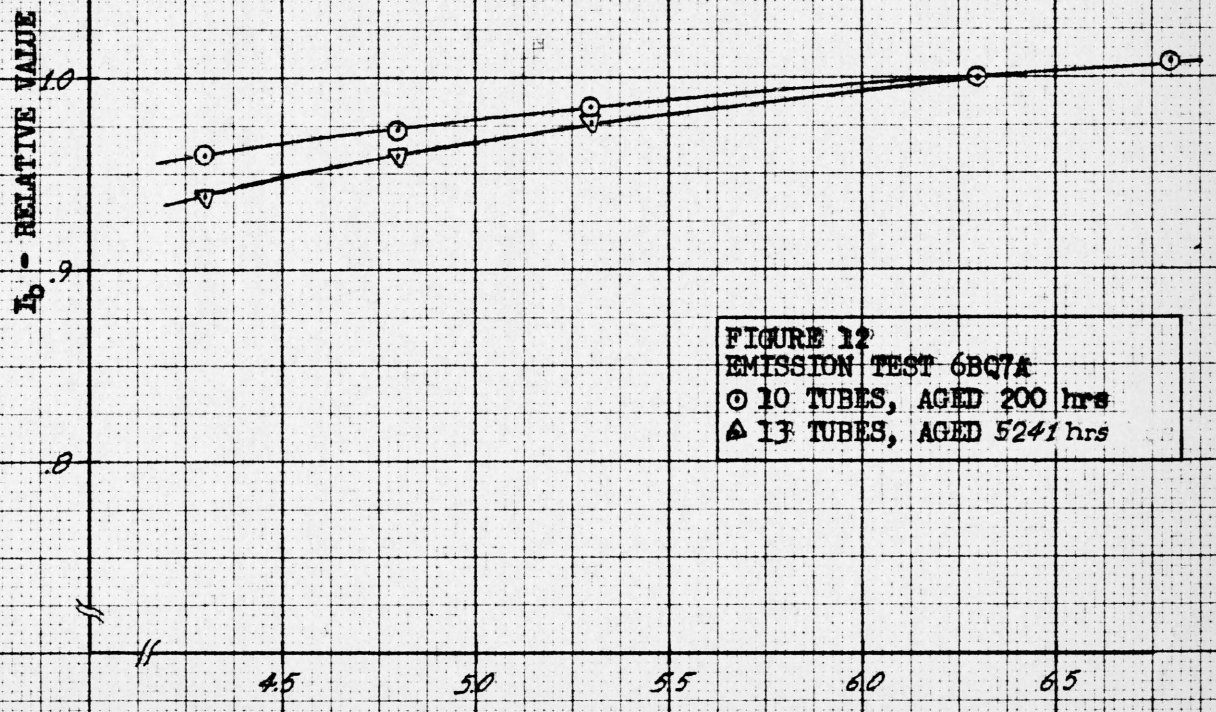


FIGURE 12
EMISSION TEST 6BQ7A
⊙ 10 TUBES, AGED 200 hrs
▲ 13 TUBES, AGED 5241 hrs

It should be noted, however, that the discussion above has been concerned with the change in I_b for an average tube while individual tubes may vary considerably from the average, as evidenced by Figure 9. The question still is not definitively answered, then, whether increased heater sensitivity for some particular tube might be indicative that this tube is rapidly approaching the end of life point. Although there is little published data available on this subject, the conclusion reached in the case of the 6AN5 tubes used in the SEAC was that there is no "connection whatsoever between filament sensitivity and an imminent decrease in plate current".[5]

From the above discussion it would appear that there is little future for heater voltage variation as a marginal checking tool. Nevertheless, this author feels that insufficient data is available at the present time to warrant the conclusion that this variable should not be investigated further. For this reason some provisions for varying heater voltage, to be discussed later, are planned for the WISC while the tube program will continue to be used to gather more information that might be used in such an investigation.

D. Subdivision

1. D.C. Supplies

The d.c. power used in the WISC is derived from two different sources: One, a pair of generators regulated against line and load changes, provides the current for the positive supplies while the other, a conventional transformer rectifier system, produces the

---

5. P. D. Shupe, Jr., and R. A. Kirsch, "SEAC - Three Years of Operation", Proceedings of the Eastern Joint Computer Conference, 1953, 90.

current required by the negative supplies. The raw output of the two generators are filtered to remove brush ripple and then subdivided into the twelve +275 volt and 24 +150 volt distribution lines which supply the computer. In addition, there are four positive supplies independently variable from +130 to +200 volts which are obtained from the raw +275 volt line by means of electronic regulators. Similarily, regulators are used to obtain the five -250 volt supplies and the one -60 volt supply from the raw B- developed by the transformer rectifier system mentioned above. These 46 d.c. supply lines, which are numbered consecutively in the order mentioned, are first fused and monitored, then distributed to the various units of the computer.

Figure 13 is a chart showing among other things the physical distribution of the main d.c. supply lines. It can be seen that each supply may be connected to several chassis; for example, supply number 42 is the -250 volt supply for all of the circuits in bays 2, 3, 4 and 5. Since the -250 volt supply for counters and flip-flops has been chosen as a marginal checking variable, this power supply line could also be used as a marginal checking line. Although the voltage on this particular marginal checking line could easily be varied at the regulator to aid in the detection of deteriorated component, this procedure would provide little assistance in finding the location of such a deteriorated component other than localizing it in bays 2, 3, 4 and 5, an area which includes about 400 counters and flip-flops. Therefore, to utilize more fully the potentialities of this particular feature of marginal checking the main power supply lines which are used for marginal checking purposes must be divided

	ΦΦ	Φ	1	2	3	4	5	6	7	8	9	10	11	12	13
A	6 CLOCK ③ CHASSIS "A" 22 CP1	6 CLOCK ② CHASSIS "B" 22 CP2	1 E ② 12 DIGIT SHIFT REGISTER ① 13 SR12	1 D ① 12 DIGIT SHIFT REGISTER ③ 13 SR12	1 A ① 12 DIGIT SHIFT REGISTER ③ 14 SR12	1 B ② 12 DIGIT SHIFT REGISTER ③ 15 SR12	1 C ② 12 DIGIT SHIFT REGISTER ③ 15 SR12	29 ARITH DIGIT ⑫ TIME SELECTOR 7 AOTS-1	27 CLOCK PULSE ⑨ CHASSIS 7 CP3			9 INPUT ⑭ OUTPUT READER 32 RD	9 DESCRIBER ⑬ 32 D	10 TAPE PUNCH ⑮ CONTROL 34 TPC	10 DESCRIBER ⑬ 35 D
B	6 MM PULSE ③ DELAY UNIT 22 MMPD	6 CLOCK ② CHASSIS "C" 22 CP3	① 13 SR12	③ 13 SR12	③ 14 SR12	③ 15 SR12	③ 15 SR12	29 ARITH DIGIT ⑫ T ME SELECTOR 7 AOTS-2	27 ⑨ 7			9 INPUT ⑮ OUTPUT WRITER 32 IOW	9 SERIALIZER ⑭ 32 SP	10 FORMAT ⑮ CONTROL 34 FC	10 PARALYER ⑬ 35 SP
C	6 MM, M ③ READER 22 RD	6 8 CHANNEL ② POWER PULSER PP 22	3 8 CHANNEL POWER PULSER PP 25	1 NEG CATH ② FOLLOWER 17 CFN	2 A & AB COINC DET 18 A-ABSC	2 24 CATHODE ④ FOLLOWERS 18 CF	3 NEG-CATH ⑥ FOLLOWERS 21 CFN	29 ARITH ⑫ ORDER TYPE DECODER 7 AOTD	27 PRECESSION ⑨ CONTROL 7 PC-2			9 COUNTER ⑬ COINC 32 PKC2	9 SERIALIZER ⑮ CONTROL 32 SPC	10 COUNTER ⑭ COINC 34 PKC2	10 SERIALIZER ⑬ PARALYZER CONTROL 38 SPC
D	6 MM_N, MM30 ① READER 22 RD	6 M PULSE ① DELAY UNIT 22 EUD	3 8 CHANNEL ③ POWER PULSER PP 25	1 EXTRACT ③ SUBTRACTOR 17 ES	2 B & ORDER ⑤ COINC DET 18 B-OSC	2 GATES ⑥ 18 MGC-2	3 EXTRACT WRITING COINC ⑤ 21 PKC	29 MINOR ⑫ ARITH CYCLE SELECTOR 7 MACS-1	27 PRECESSION ⑨ CONTROL 7 PC-3			9 COUNTER ⑭ COINC 32 PKC2	9 INPUT ⑬ TAPE READER 32 TR	10 COUNTER ⑭ COINC 34 PKC2	10 TAPE ⑬ PUNCH 35 TP
E	11 V12 & V55 ② VOLTAGE GENERATOR	11 ③	3 AUXILIARY ① EXT. CON 25 AEC	1 READ-WRITE ① TIME COINC DET 17 RWCD	1 S ④ 12 DIGIT SHIFT REGISTER ⑤ 14 SR12	2 A ADDRESS & ORDER ½ ADDERS ⑤ 18 MHA	3 EXTRACT WRITING COINC ④ 21	29 MINOR ⑪ ARITH CYCLE SELECTOR 7 MACS-2	27 SR B ⑧ OPERAND 7 SR-BB			9 COUNTER ⑮ COINC 32 PKC2	9 ENABLE VOLTAGE GEN-2 ⑭ 32 EVG	10 COUNTER ⑭ COINC 34 PKC2	10 CLOCK ⑮ CHASSIS B 35 CP-3
F	① 23 TVG	① ORDER DECODER	3 ② 25	1 R ① 12 DIGIT SHIFT REGISTER ② 16 SR12	⑤	1 O ⑤ 12 DIGIT SHIFT REGISTER ④ 16	3 ORDER TYPE DELAY & OUTPUT CHASSIS ⑥ ⑤ 21 OTD	29 MINOR ⑪ ARITH CYCLE SELECTOR 7 MACS-3	27 SR A ⑧ OPERAND 7 SR-AA1						
G	11 TIME CHECK ③ VOLTAGE GEN 23 TCVG	②	3 INPUT ③ ORDER & WRITE SEQUENCER 25		2 24 CATHODE ⑥ FOLLOWERS 18 CF			28 RESULT ⑪ SIGN COMPARE OUTPUT 7 RSCD	26 SR A ⑧ OPERAND 7 SR-AA2						
H	11 START-STOP ② CONTROL CHASSIS 23 SSC-1	③ 24 OD	3 BLOCK WRIT ② CONTROL CHASSIS 25 BWC	1 WRITING ③ TIME LOCKS 17 WTL	2 GATES ④ 18 MGC-1	2 ORDER ⑥ RECIRC 20 RS	2 RESULT RE-CIRC SW&SM MEM DEL ④ 20 SWD	⑪ 7	26 ARITH ⑧ ADDER CONTROL 7 AAC						
J	11 MANUAL CONTROL & ONE PULSER ① 23 OP-1	11 OUTPUT ① SEQUENCER 24 OSC	3 WRIT TIME ① COINCIDENCE DET 25 PKC1	4 WRITING ① CONTROL 17 WFFL	4 WRITING ⑤ CONTROL 17 WDS	2 RESULT ⑤ RECIRC 20 RS	2 READING AMPLIFIERS SELECTORS ⑥ 20 SMR	28 ARITH ⑩ RESULT 7 SR-RR-1	26 SUBTRAHEND ⑦ MINUEND GENERATOR						
K	11 COUNTER ③ COINC 23 PKC1	11 MINOR CYCLE SELECTOR ② 24 MCS	4 WRITER ③ 17 WSM	4 WRITING ② CONTROL 17 WTS	4 WRITER ⑥ 17 WSM	2 A OPERAND ④ RECIRC 20 RS	3 EXTRA A OP RECIRC ⑤ 21 CCC	28 ARITH ⑩ RESULT 7 SR-RR-2	⑦ 8 SMG						
L			4 WRITER ② 17 WSM	4 WRITER ③ 17 WSM	4 WRITER ④ 17 WSM	2 B OPERAND ④ RECIRC 20 RS	3 ④ 21	28 DUAL RECIRC READER ⑩ 7 RD	26 ARITH ⑦ ADDER SUBTRACTOR						
M			4 WRITER ① 17 WSM	4 WRITER ① 17 WSM	4 WRITER ⑤ 17 WSM	2 RECIRC WRITERS ⑤ 20 PD	3 8 CHANNEL D.C. DRIVER UNIT ⑥ 21 DU	28 DUAL RECIRC READER ⑩ 7 RD	⑦ 8 AAS						

14	15	16
39 STANDARD ⑥ MEMORY PRE-AMPLIFIERS SMRP	7 STD. MEMORY ⑤ CATHODE FOLLOWERS SMRCF	37 STD. MEM. ⑥ FF LOCKS SMFF
39 STANDARD ⑥ MEMORY PRE-AMPLIFIERS SMRP	7 STD. MEMORY ④ CATHODE FOLLOWERS SMRCF	37 STD. MEM. ⑥ FF LOCKS SMFF
		37 STD. MEM. ⑥ FF LOCKS SMFF

POWER SUPPLY DISTRIBUTION

SUPPLY 7	RDG. CPT. SMRCF
SUPPLY 37	RDG. CPT. SMRCF
SUPPLY 38	ALL RELAYS
SUPPLY 39	RDG. CPT. SMRP
SUPPLY 40	+130 VOLTS
SUPPLY 41	-65 VOLTS
SUPPLY 42	2, 3, 4, 5
SUPPLY 43	ΦΦ, Φ, 1
SUPPLY 44	6, 7, 8, 9
SUPPLY 45	10, 11, 12, 13
SUPPLY 46	SPARE

Figure 13

Issue No.	Revisions	By	Date
2		GH	JULY22,57

Capacitors in _____
Resistors in _____ ½ W, 10%
} UNLESS OTHERWISE NOTED

UNIVERSITY of WISCONSIN
Elec. Eng. Dept.— Computer Research
CHASSIS SPACE ALLOCATION AND POWER SUPPLY DISTRIBUTION

Dr. GH	Eng.	Date JULY17,57
Dwg No. W-015P-03B		

into a number of independently variable lines each serving a small number of circuits. Although this subdivision makes it easier to locate deteriorated components, a point of diminishing returns can be reached; for example, the extreme case where each circuit has its own marginal checking line would require over 2000 marginal checking lines while the control panel for such a system might well be as complex as the computer itself.

The proposed organization of the d.c. part of the marginal checking system may best be explained by the use of the diagrams in Figure 14 and 15. The portion of the marginal checking system involving the +275 volt supplies may be broken down into three parts: The Common source, the main power supply lines and the individual marginal checking lines. Variation in the +275 volt supply to any circuit may be accomplished by changing the output voltage of the source, the +275 volt generator, or by inserting a variable voltage source in series with the lines at the points indicated by the symbol ⊗. Therefore, there are three possible modes of operation for this portion of the system: The marginal checking voltage, or change in the variable, may be inserted at the generator where it will affect all circuits using this voltage (the voltage group); it may be inserted at any one of the main power supply lines where it will affect the circuits using this main supply line (the supply group); or it may be inserted into the individual marginal checking lines each of which serves a correspondingly smaller group of circuits (the line group). The breakdown of the portion of the system involving the -250 volt supplies is quite similar, the one exception being that the voltage and supply groups are identical since there is no

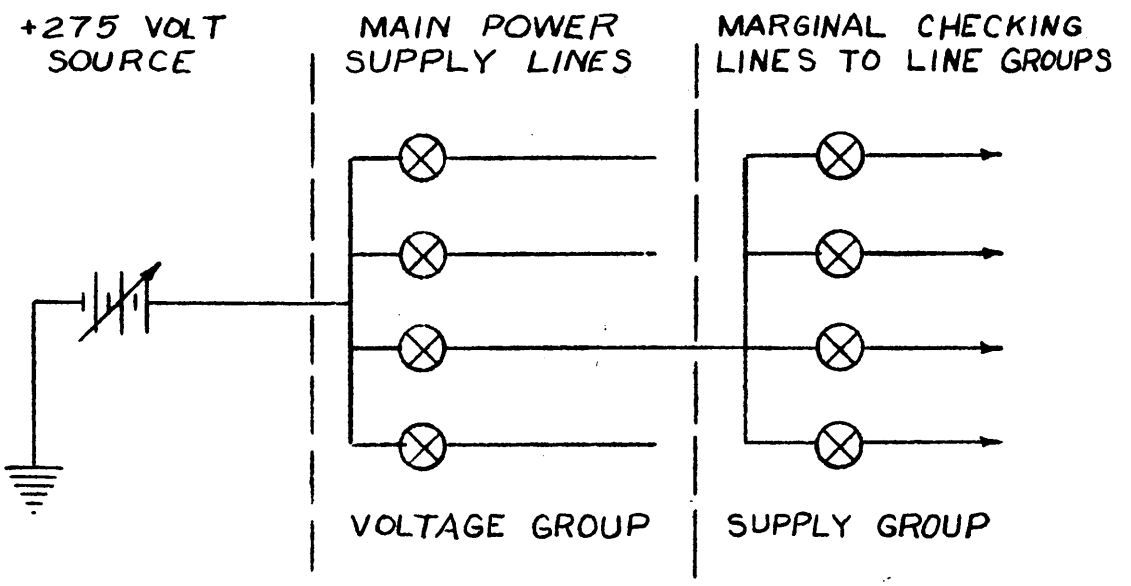


FIGURE 14
BREAKDOWN OF THE +275 VOLT MARGINAL CHECKING SYSTEM

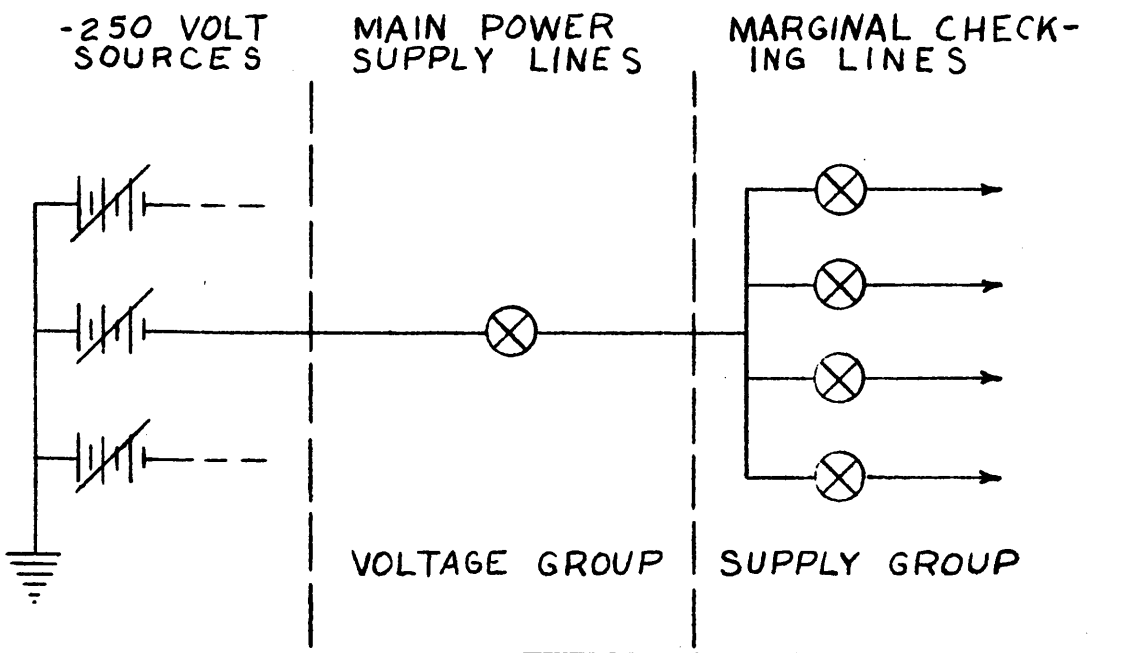


FIGURE 15
BREAKDOWN OF THE -250 VOLT MARGINAL CHECKING SYSTEM

provision for varying all of the -250 volt supplies simultaneously. As the diagrams indicate, the marginal checking voltage can be applied to one or all of the lines in a supply group; however, the system being built into the WISC will also allow the marginal checking voltage to be applied to any number of lines in the same group.

To complete the breakdown of the marginal checking system the circuits to be connected to each marginal checking line should be selected. One of the factors influencing this selection is that the margin, or magnitude of marginal checking voltage which will induce failure, for two chassis may differ considerably for several reasons although they use the same marginal checking variable. For instance, the margins of each of the three basic circuits using the +275 volt supply as a marginal checking variable--the gate, the pulse amplifier, and the cathode follower--not only are inherently different, but also depend somewhat upon the application. Consequently, it would be inadvisable deliberately to place two chassis with greatly differing margins on the same marginal checking line, for failure in the circuits with small margins will mask the prediction of errors in the circuits with larger margins. However, since the smallest marginal checking unit in the WISC is a single chassis, and since there may be as many as 50 circuits in a chassis, there will be some cases where circuits possessing quite different margins will have to be placed on the same marginal checking line. This is one of the disadvantages of having marginal checking added to the computer rather than designing the computer with marginal checking in mind.

Another factor influencing the selection of the circuits to be used on a particular marginal checking line is their logical function in the computer. For example, if one clock chassis, one standard memory chassis, one order decoder chassis, one recirculator chassis, and one arithmetic chassis were included on a particular marginal checking line, an imminent failure could be detected by the use of this line, but locating the particular component at fault would still be quite difficult because of the completely different functions of each chassis. If, however, the chassis on a particular marginal checking line were chosen so that the information being processed would pass sequentially through each, it would be a fairly simple matter to determine at which point the fault occurs unless, of course, these chassis are connected in a closed loop which cannot be opened for testing purposes.

Figure 16 is a chart similar to Figure 13 which shows the final breakdown of marginal checking lines among the chassis in the WISC. For purposes of identification, each supply group is assigned an alphabetic character while each individual line of the group is assigned a number: Thus, marginal checking line B2 is the second line in supply group B. Altogether there are 85 lines divided between 18 supply groups with each group containing from 1 to 10 lines. Although the selection of chassis for each line has been made using the ideas just discussed as a guide, the author does not claim that this breakdown is ideal or that the marginal checking system outlined here is the best that can be provided. It is hoped, however, that this system will be sufficiently useful and versatile that it will form a good foundation upon which a better system can be built in the future.

	ΦΦ	Φ	1	2	3	4	5	6	7	8	9	10	11	12	13
A	A1 ③ CLOCK CHASSIS "A" C1 CP1	② CLOCK CHASSIS "B" C1 CP2	B1 ② E 12 DIGIT SHIFT REGISTER	B1 ① D 12 DIGIT SHIFT REGISTER	B1 ① A 12 DIGIT SHIFT REGISTER	B2 ② B 12 DIGIT SHIFT REGISTER	B2 ② C 12 DIGIT SHIFT REGISTER	N1 ⑫ ARITH DIGIT TIME SELECTOR P2 ADTS-1	⑨ CLOCK PULSE CHASSIS P1 CP3			R1 ⑭ INPUT OUTPUT READER S2 RD	R4 ⑬ DESCRIBER S5 D	R8 ⑮ TAPE PUNCH CONTROL T3 TPC	R4 ⑬ DESCRIBER T5 D
B	A1 ③ MM PULSE DELAY UNIT C2 MMPD	② CLOCK CHASSIS "C" C1 CP3	① E1 SR12	③ E1 SR12	③ E1 SR12	③ E2 SR12	③ E2 SR12	N1 ⑫ ARITH DIGIT TIME SELECTOR P2 ADTS-2	⑨			R1 ⑮ INPUT OUTPUT WRITER S3 IOW	R5 SERIALIZER ⑭ S6 SP	R8 ⑮ FORMAT CONTROL T3 FC	R5 PARALYZER ⑬ T6 SP
C	A1 ③ MM, M READER C2 RD	② 8 CHANNEL POWER PULSER C1 PP	② 8 CHANNEL POWER PULSER G1 PP	B5 NEG CATH FOLLOWER ② E4 CFN	B10 A & AB COINC DET ④ F2 A-AB-SC	④ 24 CATHODE FOLLOWERS F2 CF	B5 NEG-CATH FOLLOWERS ⑥ G2 CFN	N6 ARITH ORDER TYPE DECODER ⑫ P7 AUTD	N6 PRECESSION CONTROL ⑨ P7 PC-2			R3 COUNTER COINC ⑬ S4 PKC2	R6 SERIALIZER CONTROL ⑮ S7 SPC	R3 COUNTER COINC ⑭ T2 PKC2	R6 SERIALIZER PARALYZER CONTROL ⑬ T7 SPC
D	A1 ① MM_H, MM30 READER C2 RD	A1 ① M PULSE DELAY UNIT C2 EVD	③ 8 CHANNEL POWER PULSER G1 PP	B7 EXTRACT SUBTRACTOR ③ E4 ES	B10 B & ORDER COINC DET ⑤ F2 B-O-SC	⑥ GATES F2 MGC-2	B7 EXTRACT WRITING COINC ⑤ G2 PKC	N2 MINOR ARITH CYCLE SELECTOR ⑫ P3 MACS-1	N6 PRECESSION CONTROL ⑨ P7 PC-3			R3 COUNTER COINC ⑭ S4 PKC2	R7 INPUT TAPE READER ⑬ S7 TR	R3 COUNTER COINC ⑭ T2 PKC2	R7 TAPE PUNCH ⑬ T4 TP
E	A2 VI2 & V55 ② VOLTAGE GENERATOR	A3 ③	B7 AUXILIARY EXT. CON ① G1 AEC	① READ-WRITE TIME COINC DET E4 RWCD	B3 ④ S 12 DIGIT SHIFT REGISTER E2 SR12	B10 A ADDRESS & ORDER ½ ⑤ ADDERS F3 MHA	B7 EXTRACT WRITING COINC ④ G2 PKC	N2 MINOR ARITH CYCLE SELECTOR ⑪ P3 MACS-2	N3 SR B OPERAND ⑧ P4 SR-BB			R3 COUNTER COINC ⑮ S4 PKC2	R2 ENABLE VOLTAGE GEN-2 ⑭ S1 EVG	R3 COUNTER COINC ⑭ T2 PKC2	CLOCK CHASSIS 3 ⑮ T1 CP-3
F	① D1 TVG	① ORDER DECODER	② G1	B3 R 12 DIGIT SHIFT REGISTER ① E3 SR12		B3 O 12 DIGIT SHIFT REGISTER ⑤ E3 SR12	B9 ORDER TYPE DELAY & OUTPUT CHASSIS ⑥ G3 OTD	N2 MINOR ARITH CYCLE SELECTOR ⑪ P3 MACS-3	N3 SR A OPERAND ⑧ P4 SR-AA1						
G	A2 TIME CHECK VOLTAGE GEN ③ D1 TCVG	②	A5 INPUT ORDER & WRITE SEQUENCER ③ G4 IWS	② E3 SR12	⑥ 24 CATHODE FOLLOWERS F3 CF	④ E3 SR12	⑤ G3 OTD	N7 RESULT SIGN COMPARE OUTPUT ⑪ P8 RSCO	N3 SR A OPERAND ⑧ P4 SR-AA2						
H	A4 START-STOP CONTROL CHASSIS ② D3 SSC-1	③ D2 OD	A6 BLOCK WRIT CONTROL CHASSIS ② G4 BWC	B6 WRITING TIME LOCKS ③ E3 WTL	GATES ④ F3 MGC-1	B4 ORDER RECIRC ⑥ F1 RS	B8 RESULT RE-CIRC SWITCH MEM DEL ④ F3 SWD		N8 ARITH ADDER CONTROL ⑧ P9 AAC			⑥ STANDARD MEMORY PRE-AMPLIFIERS M1 SMRP	⑤ STD. MEM. CATHODE FOLLOWERS K1 SMRCF	⑥ STD. MEM. FF LOCKS L1 SMRL	
J	A4 MANUAL CONTROL & ONE PULSER ① D4 OP-1	A5 OUTPUT SEQUENCER ① D3 OSC	A5 WRITE TIME COINC. DET ① G4 PKC1	B6 WRITING CONTROL ① H4 WFFL	B6 WRITING CONTROL ⑤ H4 WDS	B4 RESULT RECIRC ⑤ F1 RS	B5 READING AMPLIFIERS ⑥ F4 SMR	N4 ARITH RESULT ⑩ P5 SS-RR-1	N7 SUBTRAHEND MINUEND GENERATOR ⑦			⑥ STANDARD MEMORY PRE-AMPLIFIERS M2 SMRP	④ STD. MEM. CATHODE FOLLOWERS K1 SMRCF	⑥ STD. MEM. FF LOCKS L2 SMRL	
K	A7 ③ COUNTER COINC D4 PKC1	A7 ② MINOR CYCLE SELECTOR D4 MCS	WRITER ③ H3 WSM	B6 WRITING CONTROL ② H4 WTS	WRITER ⑥ H3 WSM	B4 A OPERAND RECIRC ④ F1 RS	B8 COUNTER COINCIDENCE CONTROLS ⑤ G2 CCC	N4 ARITH RESULT ⑩ P5 SS-RR-2	⑦ Q1 SMG					⑥ STD. MEM. FF LOCKS L3 SMRL	
L			WRITER ② H2 WSM	WRITER ③ H2 WSM	WRITER ④ H2 WSM	B4 B OPERAND RECIRC ④ F1 RS	B8 EXTRACT RECIRC SWITCH ④ G3 ERS	N5 DUAL RECIRC READER ⑩ P6 RD	N8 ARITH ADDER SUBTRACTOR ⑦						
M			WRITER ① H1 WSM	WRITER ① H1 WSM	WRITER ⑤ H1 WSM	RECIRC WRITERS ⑤ J1 PD	B9 8 CHANNEL D.C. DRIVER UNIT ⑥ G3 DU	N5 DUAL RECIRC READER ⑩ P6 RD	⑦ Q2 AAS						

A1
B

B⁻ MARGINAL CHECKING LINE →
FILAMENT MARGINAL CHECKING LINE
B⁺ MARGINAL CHECKING LINE →

A1 CLOCK
③ CHASSIS "A"
C1 CP1

← CHASSIS TITLE OR FUNCTION
← CHASSIS TYPE

14	15	16
⑥ STANDARD MEMORY PRE-AMPLIFIERS M1 SMRP	⑤ STD. MEM. CATHODE FOLLOWERS K1 SMRCF	⑥ STD. MEM. FF LOCKS L1 SMRL
⑥ STANDARD MEMORY PRE-AMPLIFIERS M2 SMRP	④ STD. MEM. CATHODE FOLLOWERS K1 SMRCF	⑥ STD. MEM. FF LOCKS L2 SMRL
		⑥ STD. MEM. FF LOCKS L3 SMRL

Figure 16

UNIVERSITY of WISCONSIN
Elec. Eng. Dept. — Computer Research

MARGINAL CHECKING BREAKDOWN FOR THE WISC

Dr. GH	Eng. AKS	Date AUG 9,57
Dwg No.	W-194C-01B	

2. Heater Supply

Since all of the circuits of the WISC except the negator may be marginally checked using the d.c. supply lines, it would seem that providing for heater voltage variation would be an unnecessary duplication. However, provisions for using heater voltage as a variable are being included for all circuits to allow further investigation.

At present there are fifteen heater transformers supplying the heater current for the entire computer with some heater lines carrying 80 amperes. Although the heater voltage distribution is not along logical lines in many sections of the computer, no attempt will be made to revise the wiring, since such a revision would be a large task and since the effectiveness of heater voltage variations as a marginal checking variable is still in doubt. The method to be used in the WISC to accomplish the variation is very simple: It involves nothing more than the insertion of a resistor in series with the primary of the heater transformer. In this case, however, the physical limitations which originally determined the heater voltage distribution also dictate the distribution of marginal checking lines since the two are identical. Even though the resulting marginal checking breakdown for this variable, also shown in Figure 16, is far from ideal, it should be adequate for experimental purposes.

E. Margins

Before the marginal checking system can be used regularly as a preventive maintenance tool, quantitative effects of the marginal checking voltages on the computer must be investigated. First, the failure margin or maximum variation of marginal checking voltage which can be used under

normal conditions (i.e., assuming all components are within their specified tolerances) without causing failure must be determined for each chassis, line group, supply group, and voltage group. This may be accomplished for individual chassis by examining their operation either in the computer or in special test equipment where their operation in the computer may be simulated, while the failure margins for the line, supply, and voltage groups will have to be determined using the computer alone. A record of the failure margins found in this manner should be maintained.

Once normal failure margins have been determined for the entire computer they may be utilized in two different ways for preventive maintenance. One method requires that failure margins be redetermined and recorded during each preventive maintenance period, while the other method requires only that the computer pass the preventive maintenance tests with a predetermined value of marginal checking voltage applied to each checking line. The main advantage of the second or preset margin method is the ease with which preventive maintenance can be accomplished; it is not necessary to record or analyze data, start or reset the computer many times, or vary margins. Since the purpose of marginal checking here is to determine the likelihood of the machine operating correctly until the next scheduled preventive maintenance period, the yes-no type indication given by this method would seem to suffice.

The first or failure method, however, has its advantage also. For example, there will be cases where deterioration of two components may produce a margin less than the preset value and cause a failure during preventive maintenance. If one component is checked and replaced while the other is overlooked, the performance of the circuit containing these

components might be improved to the point where the preventive maintenance tests will be passed if preset margins are used. However, the failure margin for this circuit after the replacement may be only slightly greater than the test or preset margin in which case the one component which was replaced might again require replacement during the next preventive maintenance period. Such a replacement, then, is in the nature of a temporary repair, and if the computer contains a number of these temporary repairs the reliability will undoubtedly suffer. If, however, the failure margin were determined and compared with the normal failure margin, both deteriorated components could be found. The procedure that is proposed by the author requires the preset margin method to be used for frequently scheduled preventive maintenance periods with the failure method being used for more comprehensive check-up periods. In this way, advantages of both methods can be utilized.

CHAPTER III

TEST ROUTINES

The routines to be discussed in this portion of the thesis are for use with the fixed point version of the WISC and are of the self checking type; that is, they require the computer to check its own operation and to give external evidence of its failures by either halting or printing out certain data. During the debugging stage many routines were devised to make the computer perform certain operations so that it could be checked by other external means such as indicating lights and oscilloscope displays. Although these were very useful then and will still be used as the occasion demands, they are also very time consuming compared to the more efficient self-checking routines to be considered here.

A. Reliability Routines

The purpose of a reliability routine is primarily to determine whether the machine is functioning properly at the time the routine is used; consequently,there need not be any provisions in such a routine for diagnosing any faults which might occur, or even for indicating which operation failed. However, since reliability routines are generally followed by some diagnostic procedures when they indicate a failure, it is desirable to include in some of these routines provisions for indicating the general area in which the trouble occurs.

1. Preventive Maintenance Test Routines

The major portion of the preventive maintenance to be done on the WISC will be accomplished through the use of the marginal checking facilities discussed in Chapter II while the machine is operating on the routines to be discussed here. The preventive maintenance performed in this fashion should be fairly simple so that it does

not require a great deal of computer time; yet, it must be comprehensive. Since a set of really comprehensive routines would involve reading and writing in all memory locations, it is apparent that some compromise must be reached. The solution proposed is to reserve certain periods of the operating day for the use of short yet comprehensive routines while taking time at less frequent intervals to use a set of longer and more comprehensive routines for a more exhaustive check up. The first set of routines will be presented here while the second, the acceptance test routine, will be presented later (see section III A 2 below).

a. Philosophy and General Approach

(1) Factors Affecting Operation

In its present form the WISC is designed to carry out 16 different types of orders. The manner in which the computer processes these orders, and consequently the circuits used, depends not only on the type of order called for, but also on such factors as the sequence of orders, the magnitude and signs of the operands, the storage location of words, etc. Although certain portions of the circuitry are utilized for many or all combinations of these factors, other portions are not used except under a specific combination of factors. For instance, a certain gate in the extract circuitry, gate 7a, will be used only if an extract order calls for an even shift and for the A operand to be obtained from short memory. As an example of the reasoning followed in determining the circumstances under which a particular circuit will be used, refer to Figure 17 showing the logical design of the A operand

Capacitors in _____
Resistors in _____ ½ W, 10%
UNLESS OTHERWISE NOTED

Issue No.	Revisions	By	Date

UNIVERSITY of WISCONSIN
Elec. Eng. Dept.– Computer Research

Figure 17
A Operand Recirculator.

Dr. XPX	Eng. XPX	Date

Dwg No. W-122L-01A

recirculator loop which includes this gate. The heavy line indicates the path through which information passes during the first minor memory cycle when gate 7a is to be used. It can be seen, then, that this circuit is used only during extract orders (E2) calling for the A operand from short memory (SMFF"1") in which the A operand is to be shifted an even number of digits (Single Shift "0").

Consequently, a program designed to test all of the circuits of the computer must use each order type under several sets of circumstances and must call for the use of certain features not directly associated with any specific order type.

(2) Testing Order Types

The following chart tabulates the order types along with the factors which must be included in a program designed to test thoroughly the circuitry associated with each order type. Also included is a tabulation of the combinations of factors which would be used in the preventive maintenance routines for the WISC if each order type were to be tested separately. Certain combinations of factors are unique and must be combined in one order to activate a particular circuit while the remainder of these combinations are chosen so that the number of orders required is a minimum. In the actual preventive maintenance routines used, however, the order types are not tested separately, but in a program containing many order types. Consequently, a number of these combinations, those indicated by an asterisk, will not be used in the preventive maintenance routines to follow.

ORDER TYPE	PROGRAM FACTORS	COMBINATIONS TO BE USED
Input I	MM Position In Standard Memory MM Position in Buffer Memory Buffer Memory Condition (Full-Not Full) Start External Stop Code Controls Clear Empty	See Discussion of Input-Output Routine, Section III A 1 b (1)
Extract E	X<Y (Left Shift) X>Y (Right Shift) X = Y (No Shift) Odd Shift (Greater Than 1) Short Memory of Operand A	Odd Shift, X<Y Odd Shift, X>Y Short Memory, X = Y
Multiply M	Sign of Operand A Sign of Operand B	A negative, B positive A positive, B negative* A positive, B positive (See Note 1)
Divide D	Sign of Operand A Sign of Operand B	A negative, B positive* A positive, B negative A positive, B positive* (See Note 1)
Spare		(See Note 2)
Transfer TU	Breakpoint Digit Breakpoint Switch	See Discussion of Transfer-Halt Routine, Section III A 1 b (2)
Halt H	Breakpoint Digit Breakpoint Switch	See Discussion of Transfer-Halt Routine, Section III A 1 b (2)
Output O	MM Position in Standard Memory MM Position in Buffer Memory Buffer Memory Condition (Empty - Not Empty) Format Control External Punch Stop Code Controls Tape Feed Clear Empty	See Discussion of Input-Output Routine, Section III A 1 b (2)
Add A	Sign of Operand A Sign of Operand B A>B (Absolute Values) B>A	A positive, B neg. B>A A negative, B pos. A>B A positive, B positive (See Note 3)
Add Absolute AA		A positive, B neg. B>A (See Note 3,4)
Subtract S	Sign of Operand A Sign of Operand B A>B B>A	A positive, B pos. B>A A negative, B neg. A>B A positive, B negative* (See Note 3)

ORDER TYPE	PROGRAM FACTORS	COMBINATIONS TO BE USED
Subtract Absolute SA	A > B B > A	A negative, B pos. A > B* B > A (See Note 3,4)
Transfer On Zero TZ	Sign of Operand A Sign of Operand B A = B (Result = 0) A ≠ B	A positive, B pos. A = B A negative, B neg. A ≠ B A positive, B negative* (See Note 3)
Transfer On Zero Absolute TZA	A = B A ≠ B	A negative, B pos. A = B* A ≠ B* (See Note 3,4)
Transfer On Negative TN	Sign of Operand A Sign of Operand B A > B B > A A = B	A positive, B pos. B > A A negative, B neg. A > B A negative, B neg. A = B A positive, B neg. A > B (See Note 3,5)
Transfer On Negative Absolute TNA	A > B A < B	A positive, B neg. B > A* A positive, A > B* (See Note 3,4,5)

Note 1. The mechanism which determines the sign of the result for the multiply order is also used for the divide order.

Note 2. As the functions of the spare order have not yet been assigned, it would seem that it would not be necessary to test it. However, it is included in the program to test the functions which it must now perform; for example, it should not allow any other order to be decoded, it should not allow any results to be written if it appears in the order shift register, and it should allow the order counter to progress to the next order.

Note 3. There are three different mechanisms by which the significant bits of a result are determined for the eight add, subtract, and comparison orders: The bits of the operands may be added, subtracted, or subtracted and complemented. The sign of the result for these eight orders is determined by the sign of operand A and the presence or absence of the signal which indicates that in generating the result digits the difference in the operands has been complemented.

Note 4. The four absolute order types, AA, SA, TZA, TNA, treat both operands as if they were positive regardless of their actual sign. Even though the mechanism they use to generate the significant bits of the results is the same as that used by their algebraic counterparts operating on positive

numbers, it is necessary in the preventive maintenance routines to test them with non-positive numbers to determine that the signal ALG (algebraic) is not being generated erroneously.

Note 5. Two groups of circuits using the same logic but different machine timing are used to generate the result sign. One group of circuits generates the sign for the result to be stored while the other generates the sign to be used in determining whether or not a transfer takes place. Although the circuits are almost identical, they must be tested separately. Consequently, the orders used in the preventive maintenance test routines may appear to test the result sign mechanism twice.

(3) Other Factors

The preventive maintenance test routines could be written using the orders indicated in the preceding chart, but such a test routine still would not provide a complete check on all of the circuits in the computer since, as mentioned previously, certain features of the WISC are not associated directly with any particular order type. The following discussion will be devoted to these features.

(a) Short Memory

In the WISC, if it is desired that a particular order use as an operand the result of the preceding order, this result can be obtained from what is called short memory before it is written on the drum. The mechanism which accomplishes this causes the operand recirculator switch to pick up this result from a junction in the result recirculator loop during the first minor memory cycle. At the same time the switch from the normal source of the operands, standard memory, is closed by the short memory flip-flop. Although an order calling for the A operand from short memory is used in testing the extract circuits, it is necessary to include in the preventive maintenance test routines

short memory coding of both A and B operands for arithmetic orders since the short memory circuitry used for extract orders is not identical to that used for other orders. It does not particularily matter, however, which two arithmetic orders use preceding results so long as this result is not identical with the operands which would most likely be generated if the short memory circuitry were not functioning properly. The most probably erroneous operands are zero, the operand of the preceding order, and the contents of storage location zero (when the hexadecimal short memory address 800 is used).

(b) Block Writing

The block writing control circuits (BWC) do not allow the results generated for some orders to be written. The signals which activate the BWC are d39 which is present for all spare, halt, transfer, output and compare orders and P.D. (prevent decoding) which is present for some cycles of the input, output, extract, compare, transfer and halt orders. The presence of a one in bit position twelve of any binary result address (short memory coding for the result) will also cause writing to be blocked; hence, it will be necessary to include in the preventive maintenance test routines some order with a short memory result and a test to see whether or not the result of this order was written. The action of the signals P.D. and d39 can be tested in a similar fashion. If the preventive maintenance test routines are of an iterative nature a simple transfer order will test the main portion of the BWC and d39 signal generation. Since the P.D. signal not only blocks writing, but also, as its name implies, prevents the decoding of some order that has entered the order registers, it is possible to test the generation of this signal by a proper sequence of the orders which should cause it to be present.

(c) Read-Write Coincidence

Since simultaneous reading and writing in the standard memory section of the WISC cannot be allowed, some provisions had to be included in the design to prevent this occurence. The read-write coincidence detector circuits (RWCD) compare the minor memory positions from which the A operand, B operand, and following order are to be read with the minor memory position in which the result is to be written. If a coincidence is detected, a delay one cycle signal (D1C) is generated which holds up the reading of all operands until the result has been written. Consequently, to test this feature of the machine it is necessary to ensure that the three possible coincidences do occur in the test routine.

(d) Storage Locations

Finally, the preventive maintenance performed on the machine should ideally test all available storage locations. As mentioned previously, this will be done only during the complete check-up periods using the acceptance test routines because of the large amount of time involved. However, a good approximation can be attained by having the more frequently used test routines read and write in each minor memory position and at least one minory memory position on each available standard memory track. Therefore, the preventive maintenance test routines are programmed so that the orders, operands, and results are distributed over the standard memory.

b. Proposed Routines

Although all of the circuits in the WISC could be tested by a single routine, such a routine would be extremely time consuming. For example,

while marginal checking was being performed on the arithmetic unit using such a routine, the operator would spend most of his time not adjusting the arithmetic margins, but waiting for the input and output units or manipulating the controls on the console while the program was testing the breakpoint transfer and hold operations. Consequently, three routines are used: One to test the input and output orders (the input-output routine), one to test the transfer and halt orders (the transfer-halt routine), and one to test the remainder of the machine (the general routine).

(1) The Input-Output Routine

Probably the testing of the input and output orders can best be accomplished by reading in a group of constants and then punching them out again for comparison with the original group. There are other factors to be considered, however, besides the ability of the machine to read in and punch out numbers correctly. Aside from the manual controls, three other details must be checked by this program.

First, due to the difference in speeds between the mechanical tape handling units and the remainder of the computer, two buffer or intermediate memories of 32 words each are used in the WISC. Circuits are provided in the input and output units to synchronize the operation of the order which attempts to empty (fill) this buffer memory during an input (output) order and the input (output) unit which attempts to fill (empty) it. To check these circuits the program must be designed to activate them. This is accomplished for the input unit by allowing an input order to be decoded when the input buffer memory is empty, and again by allowing the input unit to fill the buffer memory while the machine is halted. Two of the

output functions are tested by the output procedure: An output order is programmed at normal machine speed which will more than fill the buffer memory. This first causes the control unit to hang up on a "full" condition while waiting for the punch; then after the punch has cleared enough room to allow the computer to proceed, tests the ability of the control unit to terminate the output order and the ability of the punch unit to stop itself when the buffer memory has become "empty".

Second, during the processing of an input order the pertinent words in the buffer memory are written into the standard memory in the addresses specified by the order. The mechanism by which this is accomplished is not so simple as it may sound. For example, if a word stored in the buffer memory is available from the buffer memory reader unit during the sixth minor memory cycle and is to be written into standard memory during the tenth minor memory cycle, it may be transferred to the result recirculator and then to the standard memory in a single major cycle. However, if the same word in the buffer memory is to be written into the standard memory during the second minor memory cycle it will take two major cycles to accomplish this transfer since the word is not available during the first major cycle until after it is to be written. Consequently, this word will be transferred to the result recirculator during the first major cycle and written in standard memory the second. Much the same process is used for an output order except that the words are transferred from the standard memory to the output buffer memory. To test the circuits provided to deal with this situation, the storage locations used in the input-output routines are chosen so that both timing situations will arise.

Third, the design of the parallel coincidence detector circuits used in the input-output units is such that a marginal crystal diode may cause one (or more) of the input or output words being stored in the buffer memory to be written in the wrong minor memory position during the loading of the buffer memory. To test all of the possible locations, then, it is best to have at least 32 words of input and output in the test routine. This does not cause any conflict, however, since, as was previously pointed out, the buffer memories should be filled to test the synchronization between input and output units and the test of the machine.

(a) Flow Diagram and Coded Routine

A flow diagram of the input-output routine is shown in Figure 18 while the coded routine may be found on page 58. The routine may be seen to consist of two main parts, the subdivision being required for marginal checking purposes. The first part of the routine provides for four words of input and output while the second calls for 36 words. Each section of the routine is in the form of a closed loop with the shift from one to the other being accomplished by means of the breakpoint transfer switch.

The constants to be used should be chosen with two considerations in mind: First, it should be possible to check the output words for errors almost at a glance. Second, the constants should contain the sequences of binary digits which are most difficult for the buffer memory reading circuits to reproduce. Although the constants chosen do satisfy these two requirements, the author does not wish to imply that these are

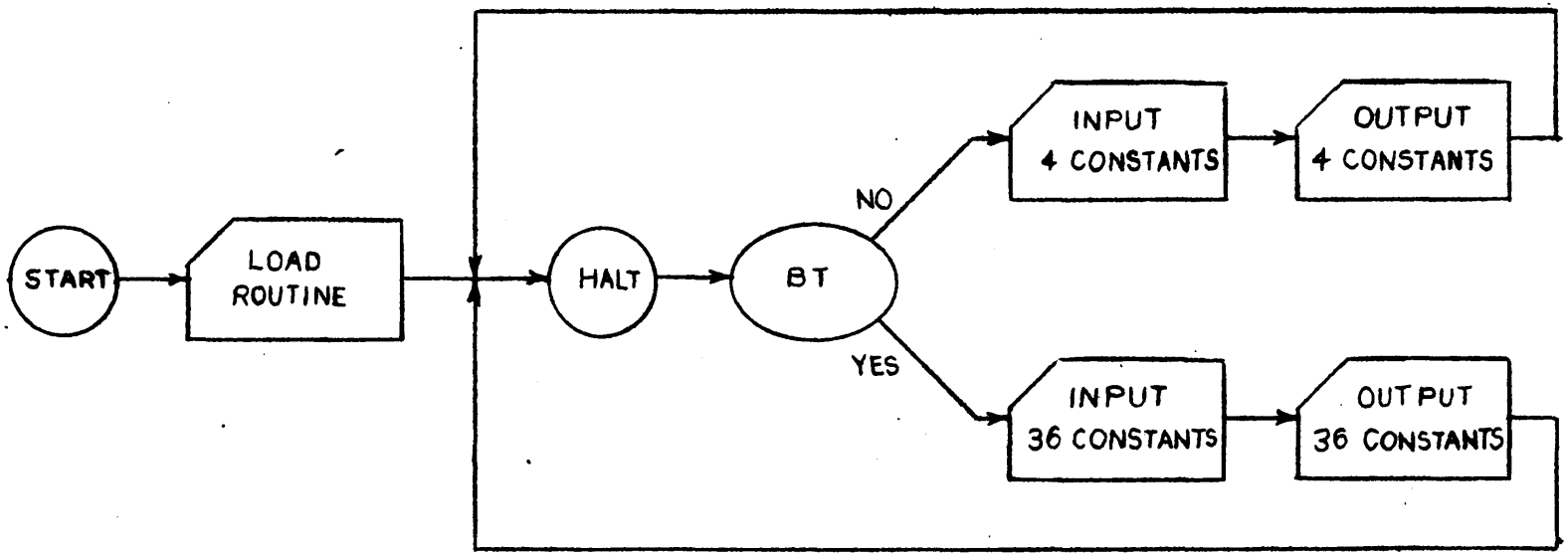


FIGURE 18
FLOW DIAGRAM OF THE INPUT-OUTPUT PREVENTIVE MAINTENANCE ROUTINE

WISCoding for PREVENTIVE MAINTENANCE INPUT-OUTPUT ROUTINE

By A. K. Scidmore Date _____ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
				[ ]	STOP CODE	[ ]						
1.00	0		I	[1]	[6]	[3]		000	0	001	006	003
1.01	1		BT	[/]	[/]	[6]		100	5	000	000	006
1.02	2		I	[8]	[11]	[4]		000	0	008	00b	004
1.03	3		H	[/]	[/]	[1]		000	6	000	000	001
1.04	4		O	[8]	[11]	[3]		000	7	008	00b	003
2.01	5		O	[8]	[47]	[3]		000	2	008	02b	003
2.02	6		I	[8]	[47]	[5]		000	1	008	02b	005
				[ ]	STOP CODE	[ ]						
				[ ]	[ ]	[ ]						
3.01	8			32 [Constants	[ ]	[ ]		001	0	fac	ade	001
3.02	9			[ ]	[ ]	[ ]		002	0	fac	ade	002
3.03	10			[ ]	[ ]	[ ]		003	0	fac	ade	003
3.04	11			[ ]	[ ]	[ ]		004	0	fac	ade	004
	•			[ ]	[ ]	[ ]						
	•			[ ]	[ ]	[ ]						
	•											
	43			[ ]	[ ]	[ ]		032	0	fac	ade	032
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

the only or even that these are necessarily the best possible constants that could be selected; however, they should suffice until experience indicates a better choice.

(b) Preventive Maintenance Procedures

This routine utilizes the marginal checking facilities as follows: First, the entire routine should be gone through without applying any marginal checking voltages to determine that the machine is running properly under normal conditions. Next, the routine should be attempted while a marginal checking voltage equal to the test value is applied to each supply group. Should the routine be passed successfully under these conditions, it then should be used while the marginal checking voltages are applied to each line group that has a test margin which is of greater magnitude than that of the supply group as a whole. If the machine passes all of these tests successfully there should be no need to continue testing the input or output orders and the next preventive maintenance routine should then be started. If the routine is <u>not</u> performed successfully while the marginal checking voltage is being applied to a supply group, the component causing the difficulty can be localized to a particular marginal checking line by applying the margin to say half of the lines in the supply group, then half of the remaining lines, etc. Once a trouble has been traced to a particular line, the source and the nature of the introduced error will have to be determined by other means. Since the input and output portions of the machine do not lend themselves very well to programmed diagnostic procedure, the final trouble shooting on the input and output circuits of the machine will have to be accomplished by the use of oscilloscopes, indicator lights, etc.

The routine is subdivided into a 4 and a 36 word test to expedite the testing procedure. To test the input and output orders thoroughly requires, as already indicated, 32 words of input and more than 32 words of output. However, certain portions of the circuitry for both input and output such as the buffer memory writers and readers, tape readers, tape punch, clock chassis, order decoder circuits, etc. can be tested adequately using only a few words. Hence, when marginal checking is being applied to these sections, the four word loop will be used, while when testing the describer and parallel counter-coincidence detector units the 36 word loop will be used.

(2) The Transfer - Halt Routine

The halt and transfer orders are unique in that they may be changed from an unconditional to a conditional nature if a one is coded in the 49th bit position of each order. Coded in this fashion, they are termed breakpoint halt and breakpoint transfer respectively, and the operation of the machine on these breakpoint orders is dependent upon the setting of a pair of switches on the console. For example, if the breakpoint transfer switch is set of "No" all breakpoint transfer orders will be ignored and the machine will decode the next sequential order while if the switch is set to "Yes" the machine will perform a normal transfer.

(a) Flow Diagram and Coded Routine

A flow diagram of the transfer-halt routine is shown in Figure 19 while the coded routine may be found on page 62. As can be seen from the block diagram, the routine consists of only four orders plus the loading order. There are three possible loops in this routine. With the breakpoint transfer switch set to "Yes" the machine will

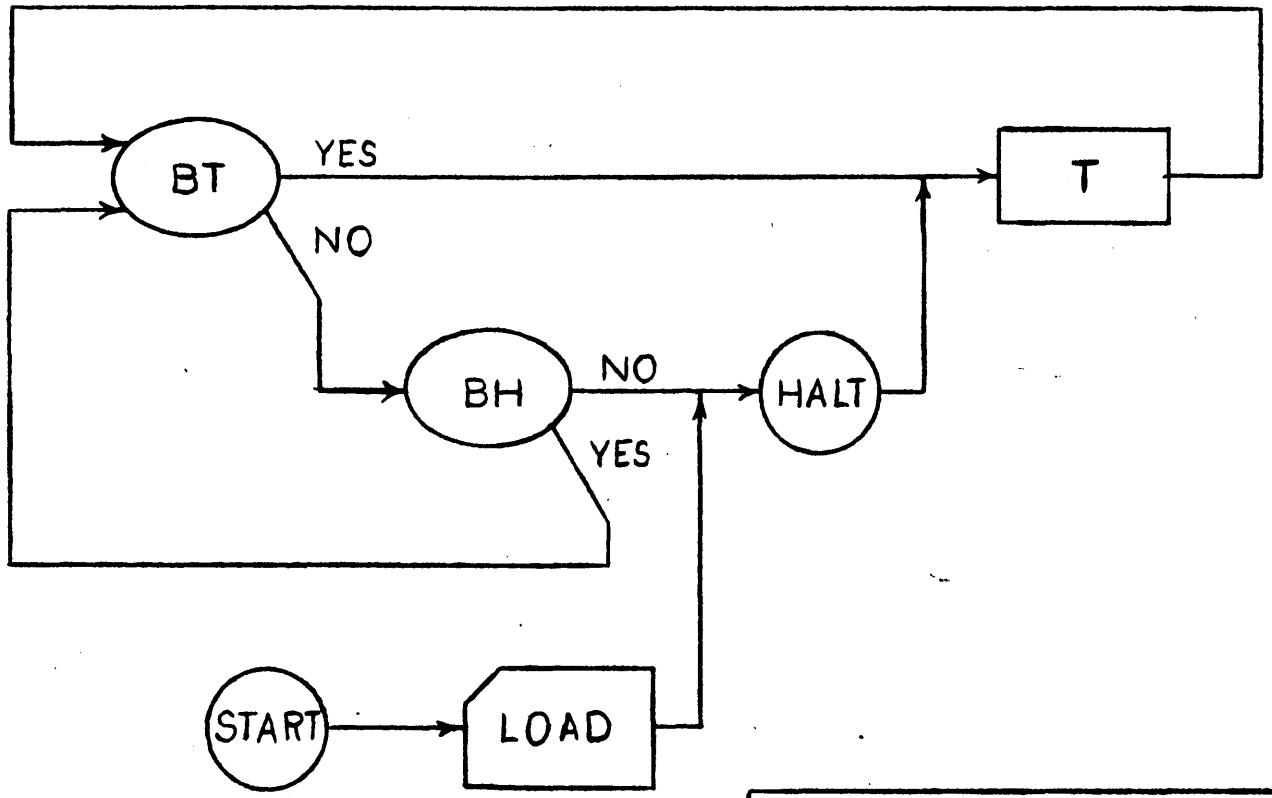


FIGURE 19
FLOW DIAGRAM OF THE TRANSFER-HALT
PREVENTIVE MAINTENANCE ROUTINE

WISCoding for PREVENTIVE MAINTENANCE TRANSFER-HALT ROUTINE

By A. K. Scidmore Date ______ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
0.00	0		I	[0]	[4]	[3]		000	0	000	004	003
				[ ]	[ ]	[ ]						
0.00	0		H	[0]	[0]	[1000]		000	6	000	000	3e8
1.00	1	1	BT	[/]	[/]	[4]		100	5	000	000	004
1.01	2	1	BH	[/]	[/]	[1]		100	6	000	000	004
1.02	3		H	[/]	[/]	[4]		000	6	000	000	004
1.03	4		T	[/]	[/]	[1]		000	5	000	000	004
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

operate in a two order loop involving only the two transfer orders. With the breakpoint transfer switch set to "No" the machine will halt because of either the breakpoint halt or the normal halt. Since after the machine has halted the order register will display the next order, the cause of the halt can be seen from the console display. For example, if the breakpoint halt switch is set at "Yes" the contents of the order register after the machine has halted will be the breakpoint transfer order.

(b) Preventive Maintenance Procedures

The marginal checking procedure to be followed here is much the same as that for the input-output routine: The routine is first used under normal conditions, then marginal checking voltages are applied. However, the only circuits to be checked by this routine are those specifically associated with these two orders, since the circuits used to locate and hold orders or produce clock pulses, though used in processing these orders, can more profitably be checked by using the general routine.

(3) The General Routine

This is the most important of the three preventive maintenance reliability routines. It is intended to be a quite comprehensive, high speed, closed loop routine, which should test all of the computer circuitry except that which is directly associated with the slow speed operations input, output, and halt. It will be assumed that these operations are adequately tested by the input-output and transfer-halt routines previously presented.

(a) Flow Diagram - Coded Routine

The routine presented here is one solution to the problem of designing a preventive maintenance test routine which attempts to incorporate all of the factors affecting the operation of the computer which were mentioned in the discussion of section III A 1 a.

A flow diagram of the routine is shown in Figure 20 while the actual coded routine appears on pages 66 through 74. The first four test blocks shown in the flow diagram incorporate all of the orders indicated previously in section III A 1 a as being necessary except the input and output orders. Consequently, the test loop consisting of test blocks 1, 2, 3 and 4 which is formed by setting the break-point transfer switch to "No" is almost a complete test routine in itself. The second half of the routine tests the ability of the machine to write in the 26 available tracks and read from all tracks. Although short memory coding, read-write coincidence, and block writing are tested thoroughly, reading and writing in <u>all</u> storage locations is not attempted. As an approximation, however, results are written during all minor memory cycles, and during at

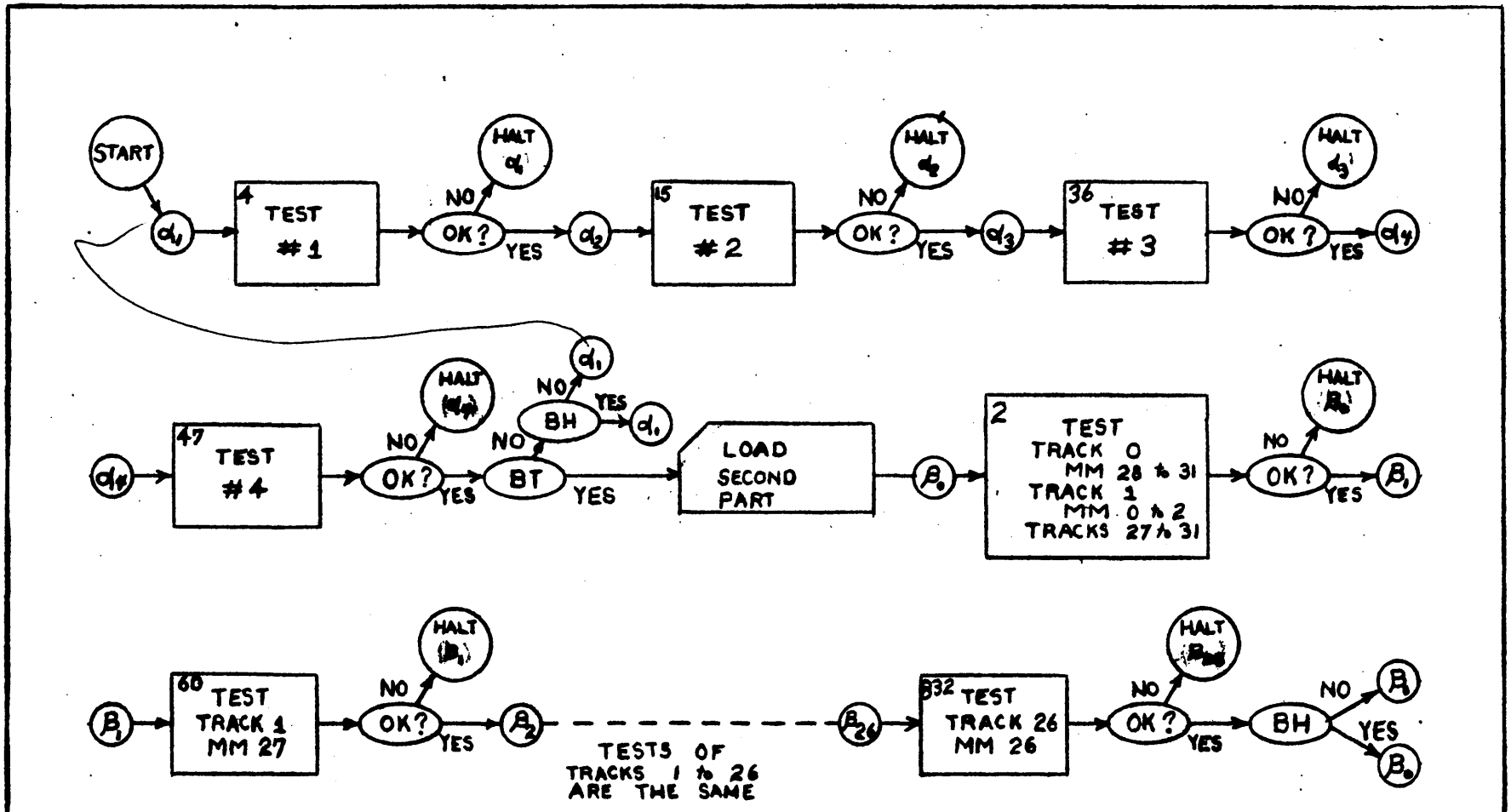


FIGURE 20
FLOW DIAGRAM OF THE GENERAL
PREVENTIVE MAINTENANCE ROUTINE

Correct Contents:

080	000 96efc 96efc	2C
081	2004b77e 4b77e	-C
082	2004b77e 4b77e	-C
083	000 96efc 96cf8	8K
084	200 25bbf25bbe	-2K
085	2003899eb899d	-3K
086	200d3a6b b6916	$C_1 C_2$
087	0000d5555555	$C_3/8C_1$
088	2005e55ddes3b	-5K

WISCoding for GENERAL PREVENTIVE MAINTENANCE ROUTINE

By A. K. Scidmore Date _____ Page 1 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
0.00	0		I	[000]	[002]	[001]		000	0	000	002	001
0.01	0		TU	[/]	[/]	[005]		000	5	000	000	004
0.02	1		I	[4]	[25]	[1023]		000	0	004	019	3ff 002
0.03	2		I	[36]	[59]	[11]		000	0	024	036	00b 03b
1.00	4		A	0[1023]+0	[1023]=0	[130]	004	000	8	3ff	3ff	082
1.01	5		AA	$+c/2$ [12] $-3c/2$	[13]=2c	[128]	5	000	9	00c	00d	080 96efc
1.02	6		A	-3c [14]+ "	[\]=-c	[129]	6	000	8	00e	800	081
1.03	7		SPARE	[13]	[13]	[128]	IGNORE 7	000	4	00d	00d	080
1.04	8		A	2c [128]+(-3c)	[14]=-c	[\]	8	000	8	080	00e	800
1.05	9	1	E	" [\]	[150]=-c	[130]	9	001	1	800	c12	082
1.06	10		TZ	-c [129]-(-c)	[130]=0	[15]	a	000	c	081	082	00f
1.07	11		H	[/]	[/]	[4]	b	000	6	000	000	004
	12			[ ] $+c/2$	[ ]	[ ]	c	000	2	5bb	f25	bbf
	13			Constants $-3c/2$	[ ]	[ ]	d	200	7	133	d71	33d
	14			[ ] $-3c$	[ ]	[ ]	e	200	e	267	ae2	67a
2.00	15		A	0[1023]+0	[1023]=0	[131]	0f	000	8	3ff	3ff	083
2.01	16	1	E	[22]	[447]=+8K	[131]	010	001	1	016	84f	083
2.02	17		A	-K [23]+(-K)	[23]=-2K	[132]	11	000	8	017	017	084
2.03	18		S	5K [24]-8K	[131]=-3K	[133]	12	000	a	018	083	085
2.04	19		S	-5K [25]-(-2K)	[132]=-3K	[\]	13	000	a	019	084	800
2.05	20		TZ	-3K [133]-(-3K)	[\]=0	[36]	14	000	c	085	800	024
2.06	21		H	[/]	[/]	[15]	15	000	6	000	000	00f
	22			[ ] K	[ ]	[ ]	16	000	1	2dd	f92	ddf
	23			Constants -K	[ ]	[ ]	17	200	1	2dd	f92	ddf
	24			[ ] +5K	[ ]	[ ]	18	000	5	e55	dde	55b

WISCoding for _____

By _____ Date _____ Page 2 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	25			CONSTANTS -5K [ ]	[ ]	[ ]	029	200	5	e55	dde	55b
3.00	36		A	0 [1023] + 0	[1023] = 0	[135]	024	000	8	3ff	3ff	135
3.01	37		M	C_1 [43] × C_2	[44] = C_1C_2	[134]	025	000	2	02b	02c	086
3.02	38		N	C_2 [44] × C_3	[45] = C_2C_3	[ ]	026	000	2	02c	02d	800
3.03	39		D	" [ ] ÷ C_1C_2	[134] = C_3/C_1	[ ]	027	000	3	800	086	800
3.04	40	4	E	" [ ]	[147] = $C_3/8C_1$	[135]	028	004	1	800	81f	087
3.05	41		TZ	" [135] − C_4	[46] = 0	[47]	029	000	c	087	02e	02f
3.06	42		H	[ ]	[ ]	[36]	02a	000	6	000	000	024
	43			[ ]	C_1 [ ]	[ ]	02b	000	e	267	ae2	67a
	44			CONSTANTS	C_2 [ ]	[ ]	02c	200	e	f56	feb	7cb
	45			[ ]	C_3 [ ]	[ ]	02d	000	5	e55	dde	55e
	46			[ ]	$C_3/8C_1$ [ ]	[ ]	02e	000	0	d55	555	555
4.00	47		A	0 [1023] + (-5K) [25]	= -5K	[136]	02f	000	8	3ff	019	088
4.01	48		TN	K [22] − 5K	[24] = -4K	[50]	030	000	e	016	018	030
4.02	49		H	[ ]	[ ]	[47]	031	000	6	000	000	02f
4.03	50		TN	-5K [25] − (-K)	[23] = -4K	[52]		000	e	019	017	034
4.04	51		H	[ ]	[ ]	[47]		000	6	000	000	02f
4.05	52		TN	-5K [25] − (-5K)	[25] = 0	[58]		000	e	019	019	03a
4.06	53		TN	K [22] − (-5K)	[25] = +4K	[58]	035	000	e	016	019	03a
4.07	54		TZ	K [22] − (-5K)	[36] = +4K	[5]	036	000	c	016	088	03a
4.08	55		BT	[ ]	[ ]	[59]		100	5	000	000	03b
4.09	56		BH	[ ]	[ ]	[4]	1→	000	5	000	000	004
4.10	57		T	[ ]	[ ]	[4]		000	6	000	000	004
4.11	58		H	[ ]	[ ]	[47]	1→	100	6	000	000	02f
4.12	59		H	[ ]	[ ]	[1023]	03b	000	6	000	000	3ff

WISCoding for _____

By _____ Date _____ Page 3 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	0		I	[70]	[85]	[70]		000	0	046	055	046
	70		I	[2]	[25]	[71]		000	0	002	019	047
	71		I	[35]	[50]	[72]		000	0	023	032	048
	72		I	[60]	[67]	[73]		000	0	03c	043	049
	73		I	[124]	[131]	[74]		000	0	07c	083	04a
	74		I	[188]	[195]	[75]		000	0	0bc	0c3	04b
	75		I	[252]	[259]	[76]		000	0	0fc	103	04c
	76		I	[316]	[323]	[77]		000	0	13c	143	04d
	77		I	[380]	[387]	[78]		000	0	17c	183	04e
	78		I	[444]	[451]	[79]		000	0	1bc	1c3	04f
	79		I	[508]	[515]	[80]		000	0	1fc	203	050
	80		I	[572]	[579]	[81]		000	0	23c	243	051
	81		I	[636]	[643]	[82]		000	0	27c	283	052
	82		I	[700]	[707]	[83]		000	0	2bc	2c3	053
	83		I	[764]	[771]	[84]		000	0	2fc	303	054
	84		I	[828]	[837]	[85]		000	0	33c	345	055
	85		II	[/]	[/]	[2]		000	6	000	000	002
0.00	2		S	[0] $-ln\epsilon^2$	[1007] $=K_0$	[0]		000	a	000	3ef	000
0.01	3		S	" [\ ] $-ln\epsilon^2$	[1007] $=K_1$	[1]		000	a	800	3ef	001
0.02	4		TZ	1 [1006] $-ln\epsilon^2$	[1007] $\neq 0$	[6]	TEST	000	c	3ee	3ef	006
0.03	5		TZ	$ln\epsilon^2$ [1007] $-ln\epsilon^2$	[1007] $=0$	[7]	TRACK	000	c	3ef	3ef	007
0.04	6		II	[/]	[/]	[4]	31	000	6	000	000	004
0.05	7		TZ	[992]	[994] $\neq 0$	[9]	TEST	000	c	3e0	3e2	009
0.06	8		TZ	[994]	[994] $=0$	[10]	TRACK	000	c	3e2	3e2	00a
0.07	9		II	[/]	[/]	[7]	30	000	6	000	000	007

WISCoding for _____

By _____ Date _____ Page 4 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
0.08	10		TZ	[928]	[929] ≠ 0	[12]	TEST	000	c	3a0	3a1	00c
0.09	11		TZ	[929]	[929] = 0	[13]	TRACK	000	c	3a1	3a1	00
0.10	12		H	[ ]	[ ]	[10]	29	000	6	000	000	00a
0.11	13		TZ	[896]	[897] ≠ 0	[15]	TEST	000	c	380	381	00f
0.12	14		TZ	[897]	[897] = 0	[16]	TRACK	000	c	381	381	010
0.13	15		H	[ ]	[ ]	[13]	28	000	6	000	000	00d
0.14	16		TZ	[864]	[865] ≠ 0	[18]	TEST	000	c	360	361	012
0.15	17		TZ	[865]	[865] = 0	[19]	TRACK	000	c	361	361	013
0.16	18		H	[ ]	[ ]	[16]	27	000	6	000	000	010
0.17	19		A	0 [1023] + K_0	[0] = K_0	[28]		000	8	3ff	000	01c
0.18	20		A	K_0 [0] + 0	[1023] = K_0	[29]		000	8	000	3ff	01d
0.19	21		A	0 [1023] + K_0	[0] = K_0	[30]		000	8	3ff	000	01e
0.20	22		A	K_0 [0] + 0	[1023] = K_0	[31]		000	8	000	3ff	01f
0.21	23		A	0 [1023] + K_0	[0] = K_0	[32]		000	8	3ff	000	020
0.22	24		A	K_0 [0] + 0	[1023] = K_0	[33]		000	8	000	3ff	021
0.23	25		A	0 [1023] + K_0	[0] = K_0	[34]		000	8	3ff	000	022
0.24	35		TZ	[4]	[5] ≠ 0	[37]		000	c	004	005	025
0.25	36		TZ	K_0 [28] − K_0	[0] = 0	[38]		000	c	01c	000	026
0.26	37		H	[ ]	[ ]	[35]		000	6	000	000	023
0.27	38		TZ	K_0 [29] − K_0	[0] = 0	[40]	TRACK 0	000	c	01d	000	028
0.28	39		H	[ ]	[ ]	[38]	MMPOS. 28,29	000	6	000	000	026
0.29	40		TZ	K_0 [30] − K_0	[ ] = 0	[42]	30,31	000	c	01e	000	02a
0.30	41		H	[ ]	[ ]	[40]		000	6	000	000	028
0.31	42		TZ	K_0 [31] − K_0	[ ] = 0	[44]		000	c	01f	000	02c
0.32	43		H	[ ]	[ ]	[42]		000	6	000	000	02a

WISCoding for _____

By _____ Date _____ Page 5 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
1.00	44		TZ	[44]	[45]≠0	[46]		000	c	02c	02d	02e
1.01	45		TZ	K_o[32]-K_o	[0]=0	[47]		000	c	020	000	02f
1.02	46		H	[ ]	[ ]	[45]	TRACK 0	000	6	000	000	02d
1.03	47		TZ	K_o[33]-K_o	[0]=0	[49]	MM POS 28,29	000	c	021	000	031
1.04	48		H	[ ]	[ ]	[47]	30,31	000	6	000	000	02f
1.05	49		TZ	K_o[34]-K_o	[0]=0	[60]		000	c	022	000	03c
1.06	50		H	[ ]	[ ]	[49]		000	6	000	000	031
1.07	60		A	K_o[0]+0	[1023]=K_o	[59]		000	8	000	3ff	03b
1.08	61		TZ	[61]	[62]≠0	[63]	TRACK 1	000	c	03d	03e	03f
1.09	62		TZ	K_o[59]-K_o	[59]=0	[64]	MM 27	000	c	03b	03b	040
1.10	63		H	[ ]	[ ]	[60]		000	6	000	000	03c
2.00	64		A	K_o[0]+0	[1023]=K_o	[69]		000	8	000	3ff	045
2.01	65		TZ	[65]	[66]≠0	[67]	TRACK 2	000	6	041	042	043
2.02	66		TZ	K_o[69]-K_o	[69]=0	[124]	MM 5	000	c	045	045	07c
2.03	67		H	[ ]	[ ]	[64]		000	6	000	000	040
3.00	124		A	K_o[0]+0	[1023]=K_o	[99]		000	8	000	3ff	063
3.01	125		TZ	[125]	[126]≠0	[127]	TRACK 3	000	c	07d	07e	07f
3.02	126		TZ	K_o[99]-K_o	[99]=0	[128]	MM 3	000	c	063	063	080
3.03	127		H	[ ]	[ ]	[124]		000	6	000	000	07c
4.00	128		A	0 [1023]+K_o	[0]=K_o	[132]		000	8	3ff	000	084
4.01	129		TZ	[129]	[130]≠0	[131]	TRACK 4	000	c	081	082	083
4.02	130		TZ	K_o[132]-K_o	[132]=0	[188]	MM 4	000	c	084	084	0bc
4.03	131		H	[ ]	[ ]	[128]		000	6	000	000	080
5.00	188		A	K_o[0]+0	[1023]=K_o	[162]	TRACK 5	000	8	000	3ff	0a2
5.01	189		TZ	[189]	[190]≠0	[191]	MM 2	000	c	0bd	0be	0bf

WISCoding for _____

By _____ Date _____ Page 6 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
5.02	190		TZ	K_o[162] - K_o	[162] = 0	[192]	TRACK 5	000	c	162	0a2	0c0
5.03	191		H	[ ]	[ ]	[188]	MM2	000	6	000	000	0bc
6.00	192		A	0 [1023] + K_o	[0] = K_o	[198]		000	8	3ff	000	0c6
6.01	193		TZ	[193]	[194] ≠ 0	[195]	TRACK 6	000	c	0c1	0c2	0c3
6.02	194		TZ	K_o[198] - K_o	[198] = 0	[252]	MM6	000	c	0c6	0c6	0fc
6.03	195		H	[ ]	[ ]	[192]		000	6	000	000	0c0
7.01	252		A	K_o[0] + 0	[1023] = K_o	[231]		000	8	000	3ff	0e7
7.02	253		TZ	[253]	[254] ≠ 0	[255]	TRACK 7	000	c	0fd	0fe	0ff
7.03	254		TZ	K_o[231] - K_o	[231] = 0	[256]	MM7	000	6	0e7	0e7	100
7.04	255		H	[ ]	[ ]	[252]		000	6	000	000	0fc
8.01	256		A	0 [1023] + K_o	[0] = K_o	[264]		000	8	3ff	000	108
8.02	257		TZ	[257]	[258] ≠ 0	[259]	TRACK 8	000	c	101	102	103
8.03	258		TZ	K_o[264] - K_o	[264] = 0	[316]	M.M8	000	c	108	108	13c
8.04	259		H	[ ]	[ ]	[256]		000	6	000	000	100
9.01	316		A	K_o[0] + 0	[1023] = K_o	[297]		000	8	000	3ff	129
9.02	317		TZ	[317]	[318] ≠ 0	[319]	TRACK 9	000	c	13d	13e	13f
9.03	318		TZ	K_o[297] - K_o	[297] = 0	[320]	MM9	000	c	129	129	140
9.04	319		H	[ ]	[ ]	[316]		000	6	000	000	13c
10.01	320		A	0 [1023] + K_o	[0] = K_o	[330]		000	8	3ff	000	14a
10.02	321		TZ	[321]	[322] ≠ 0	[323]	TRACK 10	000	c	141	142	143
10.03	322		TZ	K_o[330] - K_o	[330] = 0	[380]	MM10	000	c	142	142	17c
10.04	323		H	[ ]	[ ]	[320]		000	6	000	000	140
11.01	380		A	K_o[0] + 0	[1023] = K_o	[363]		000	8	000	3ff	16b
11.02	381		TZ	[381]	[382] ≠ 0	[383]	TRACK 11	000	c	17d	17e	17f
11.03	382		TZ	K_o[363] - K_o	[363] = 0	[384]	MM11	000	c	16b	16b	180

WISCoding for _____

By _____ Date _____ Page 7 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
11.04	383		H	[/]	[/]	[380]	TRACK 11 MM11	000	6	000	000	17c
12.01	384		A	O[1023]+K₀	[0]=K₀	[396]		000	8	3ff	000	18c
12.02	385		TZ	[385]	[386]≠0	[387]	TRACK 12	000	c	181	182	183
12.03	386		TZ	K₀[396]-K₀	[396]=0	[444]	MM12	000	c	18c	18c	1bc
12.04	387		H	[/]	[/]	[384]		000	6	000	000	180
13.01	444		A	K₀[0]+O	[1023]=K₀	[429]		000	8	000	3ff	1ad
13.02	445		TZ	[445]	[446]≠0	[447]	TRACK 13	000	c	1bc	1be	1bf
13.03	446		TZ	K₀[429]-K₀	[429]=0	[448]	MM13	000	c	1ad	1ad	1c0
13.04	447		H	[/]	[/]	[444]		000	6	000	000	1bc
14.01	448		A	O[1023]+K₀	[0]=K₀	[462]		000	8	3ff	000	1ce
14.02	449		TZ	[449]	[450]≠0	[451]	TRACK 14	000	c	1d1	1c2	1c3
14.03	450		TZ	K₀[462]-K₀	[462]=0	[508]	MM14	000	c	1ce	1ce	1fc
14.04	451		H	[/]	[/]	[448]		000	6	000	000	1c0
15.01	508		A	K₀[0]+O	[1023]=K₀	[495]		000	8	000	3ff	1ef
15.02	509		TZ	[509]	[510]≠0	[511]	TRACK 15	000	c	1fd	1fe	1fg
15.03	510		TZ	K₀[495]-K₀	[495]=0	[512]	MM15	000	c	1ef	1ef	200
15.04	511		H	[/]	[/]	[508]		000	6	000	000	1fc
16.01	512		A	O[1023]+K₀	[0]=K₀	[528]		000	8	eff	000	210
16.02	513		TZ	[513]	[514]≠0	[515]	TRACK 16	000	c	201	202	203
16.03	514		TZ	K₀[528]-K₀	[528]=0	[572]	MM16	000	c	210	210	23c
16.04	515		H	[/]	[/]	[512]		000	6	000	000	200
17.01	572		A	K₀[0]+O	[1023]=K₀	[561]		000	8	000	3ff	231
17.02	573		TZ	[573]	[574]≠0	[575]	TRACK 17	000	c	23d	23e	23f
17.03	574		TZ	K₀[561]-K₀	[561]=0	[576]	MM17	000	c	231	231	240
17.04	575		H	[/]	[/]	[572]		000	6	000	000	23c

WISCoding for _____

By _____ Date _____ Page 8 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
18.01	576		A	0 [1023] + K_0	[0] = K_0	[594]		000	8	3ff	000	252
18.02	577		TZ	[577]	[578] ≠ 0	[579]	TRACK 18	000	c	2a1	2a2	2a3
18.03	578		TZ	K_0[594] - K_0	[594] = 0	[636]	MM18	000	c	252	252	27c
18.04	579		H	[/]	[/]	[576]		000	6	000	000	240
19.01	636		A	K_0[0] + 0	[1023] = K_0	[627]		000	8	000	3ff	273
19.02	637		TZ	[637]	[638] ≠ 0	[639]	TRACK 19	000	c	27d	27e	27f
19.03	638		TZ	K_0[627] - K_0	[627] = 0	[640]	MM19	000	c	273	273	280
19.04	639		H	[/]	[/]	[636]		000	6	000	000	27c
20.01	640		A	0 [1023] + K_0	[0] = K_0	[660]		000	8	3ff	000	294
20.02	641		TZ	[641]	[642] ≠ 0	[643]	TRACK 20	000	c	281	282	283
20.03	642		TZ	K_0[660] - K_0	[660] = 0	[700]	MM20	000	c	294	294	2bc
20.04	643		H	[/]	[/]	[640]		000	6	000	000	280
21.01	700		A	K_0[0] + 0	[1023] = K_0	[693]		000	8	000	3ff	2b5
21.02	701		TZ	[701]	[702] ≠ 0	[703]	TRACK 21	000	c	2bd	2be	2bf
21.03	702		TZ	K_0[693] - K_0	[693] = 0	[704]	MM21	000	c	2b5	2b5	2c0
21.04	703		H	[/]	[/]	[700]		000	6	000	000	2bc
22.01	704		A	0 [1023] + K_0	[0] = K_0	[726]		000	8	3ff	000	2b6
22.02	705		TZ	[705]	[706] ≠ 0	[707]	TRACK 22	000	c	2c1	2c2	2c3
22.03	706		TZ	K_0[726] - K_0	[726] = 0	[764]	MM22	000	c	2d6	2db	2fc
22.04	707		H	[/]	[/]	[704]		000	6	000	000	2c0
23.01	764		A	K_0[0] + 0	[1023] = K_0	[759]		000	8	000	3ff	2f7
23.02	765		TZ	[765]	[766] ≠ 0	[767]	TRACK 23	000	c	2fd	2fe	2ff
23.03	766		TZ	K_0[759] - K_0	[759] = 0	[768]	MM23	000	c	2f7	2f7	300
23.04	767		H	[/]	[/]	[764]		000	6	000	000	2fc
24.01	768		A	0 [1023] + K_0	[0] = K_0	[792]	TRACK 24 MM24	000	8	3ff	000	318

WISCoding for _____

By _____ Date _____ Page 9 of 9

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
24.02	769		TZ	[769]	[770] ≠ 0	[771]	TRACK 29	000	e	301	302	303
24.03	770		TZ	K_0[792] − K_0	[792] = 0	[828]	MM24	000	e	318	318	33e
24.04	771		H	[/]	[/]	[768]		000	6	000	000	300
25.01	828		A	K_0[0] + 0	[1023] = K_0	[825]		000	8	000	3ff	339
25.02	829		TZ	[829]	[830] ≠ 0	[831]	TRACK 25	000	e	33d	33e	33f
25.03	830		TZ	K_0[825] − K_0	[825] = 0	[832]	MM25	000	e	339	339	340
25.04	831		H	[/]	[/]	[828]		000	6	000	000	33e
26.01	832		A	0 [1026] + K_0	[0] = K_0	[858]		000	8	3ff	000	35a
26.02	833		TZ	[833]	[834] ≠ 0	[835]	TRACK 26	000	e	341	342	343
26.03	834		TZ	K_0[858] − K_0	[858] = 0	[836]	MM26	000	e	35a	35a	344
26.04	835		H	[/]	[/]	[832]		000	6	000	000	340
26.05	836		HH	[/]	[/]	[2]		000	6	000	000	002
26.06	837		T	[/]	[/]	[2]		000	5	000	000	002
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

least one minor memory cycle on each available track[1], while orders and operands are read from all minor memory positions and all tracks.

(b) Preventive Maintenance Procedure

For use in preventive maintenance this routine is divided into two parts. The first part is intended to be used as a closed loop (breakpoint transfer switch set on "No") which can be performed once in about 0.75 seconds.

This portion of the routine is arranged so that the orders are stored on tracks zero and one while its results will be stored on track four, permitting the writing circuits for tracks zero and one to be disabled once the orders have been loaded. This can save much wasted time in the course of testing the machine, for a failure in the control section of the computer while it is operating at high speed may allow writing almost anywhere throughout the memory before the machine will halt or hang up, possibly destroying the stored program being used for the test. Should this occur, it is not necessary to reload the program but only to clear the machine as if to load; the order located in storage location zero (a transfer order) will restore control of the machine to the routine.

The second part of the routine which is intended primarily for testing the ability of the machine to read and write from all available tracks and all minor memory positions is also in the form of a closed loop. This portion of the general routine is distributed over the memory with at least four orders stored on each available track. After

---

1. Since the storage locations on tracks 27 through 31 are reserved for permanent storage of the more important subroutines and constants, writing in these storage locations is not permitted. Hence, this routine reads from these tracks but does not write on them.

the results for these orders have been written once all writers may be disabled while the reading circuits are checked so that a failure in reading an order cannot destroy the program. Thus, the only time it is necessary to allow writing is when the writing circuits themselves are being checked; this should considerably reduce the time required to perform the marginal checking.

The marginal checking procedure to be followed here is much the same as that for the input-output routine: The routine will be used first under normal conditions, then with marginal checking voltages being applied successively at the voltage group level, at the supply group level, and finally at the line group level. The sequence in which the various sections of the computer are tested is not critical, but some systematic procedure should be used. Work sheets will show the sequence of testing along with the normal failure margins and test margins indicated for each voltage, supply, and line group. Initially when failure margins are still being determined and during the more thorough checkups these work sheets will also be used to record failure margins.

2. The Acceptance Test Routine

a. Philosophy and General Approach

A digital computer or computing system which is produced for sale or rental to some particular user generally must pass some specified acceptance tests before it will be accepted by the user. Quite often these tests are in the form of a very comprehensive program which tests the ability of the machine to perform its designated functions for extended periods of time. In the field of digital computers the WISC is unique in that its design and construction were not undertaken

primarily to provide a computing tool for solving problems, but with objective in mind of conducting "a modest program of education, research, and construction, and to make the most of the opportunities for training young electrical engineers and acquainting them with some of the basic problems in this rapidly expanding area".[2] Consequently, there are no specific acceptance specifications to be met other than those which outline the overall logical design of the WISC. However, in order to mark a milestone in the progress of this project, the author wishes to propose the following test routine as an acceptance test for the fixed point version of the WISC. A very similar routine operated successfully for a two hour period early in the summer of 1957 so that it is hoped that the Acceptance Test Routine proposed here will be little more than a formal hurdle. This test routine, it is noted, has a two-fold purpose since, as mentioned previously, it will also be used to provide a very comprehensive test for preventive maintenance purposes.

For the purposes outlined above this routine should involve all possible combinations of factors which affect the operation of the machine. These factors were discussed in section III A 1 a, and routines which incorporate most of these factors, the Preventive Maintenance Test Routines, have already been presented. However, they do have some shortcomings: They do not read and write in all available storage locations nor do they use all possible combinations of numbers. Although the acceptance test routine can be designed to

---

2. J. L. Asmuth, C. H. Davidson, J. B. Miller, D. S. Noble, and A. K. Scidmore, "The Wisconsin Integrally Synchronized Computer -- A University Research Project", Communications and Electronics, 25 (July 1956), 330.

overcome the first of these shortcomings, it is not practical to design it to operate on all possible combinations of numbers. One solution sometimes proposed is to have the machine operate on random numbers while another is to choose the constants which are the most difficult for the machine to operate with. The scheme used here is similar to the second solution, but because of the method used to bring about writing and reading in all memory locations some of the orders in the routine may operate on as many as 300,000 different operands. This, then, could be considered as approximating the use of all possible numbers.

b. The Routine

The Acceptance Test Routine consists of two major parts: the test portion which is very similar to part of the preventime maintenance routine, and the relocation portion which transfers all of the words of the test routine from one block of storage locations to another each time the tests have been successfully completed. The latter is very similar to the adaptation routines which are used to adapt a library subroutine to the desired location in some program[3].

(1) The Test

From the flow diagram of the acceptance test routine, Figure 21, it can be seen that the test portion of this routine (tests 1 through 4) is very similar to the first four tests of the General Preventive Maintenance Routine, and an examination of the coded

---

3. O. R. Clement, "A Basic Routine Library and Programming Manual for the WISC", M.S. Thesis, University of Wisconsin, 1957:

C. H. Davidson, "Mathematical Procedures and Testing Methods for an Intermediate Speed Digital Computer", Ph.D. Thesis, University of Wisconsin, 1952, 73.

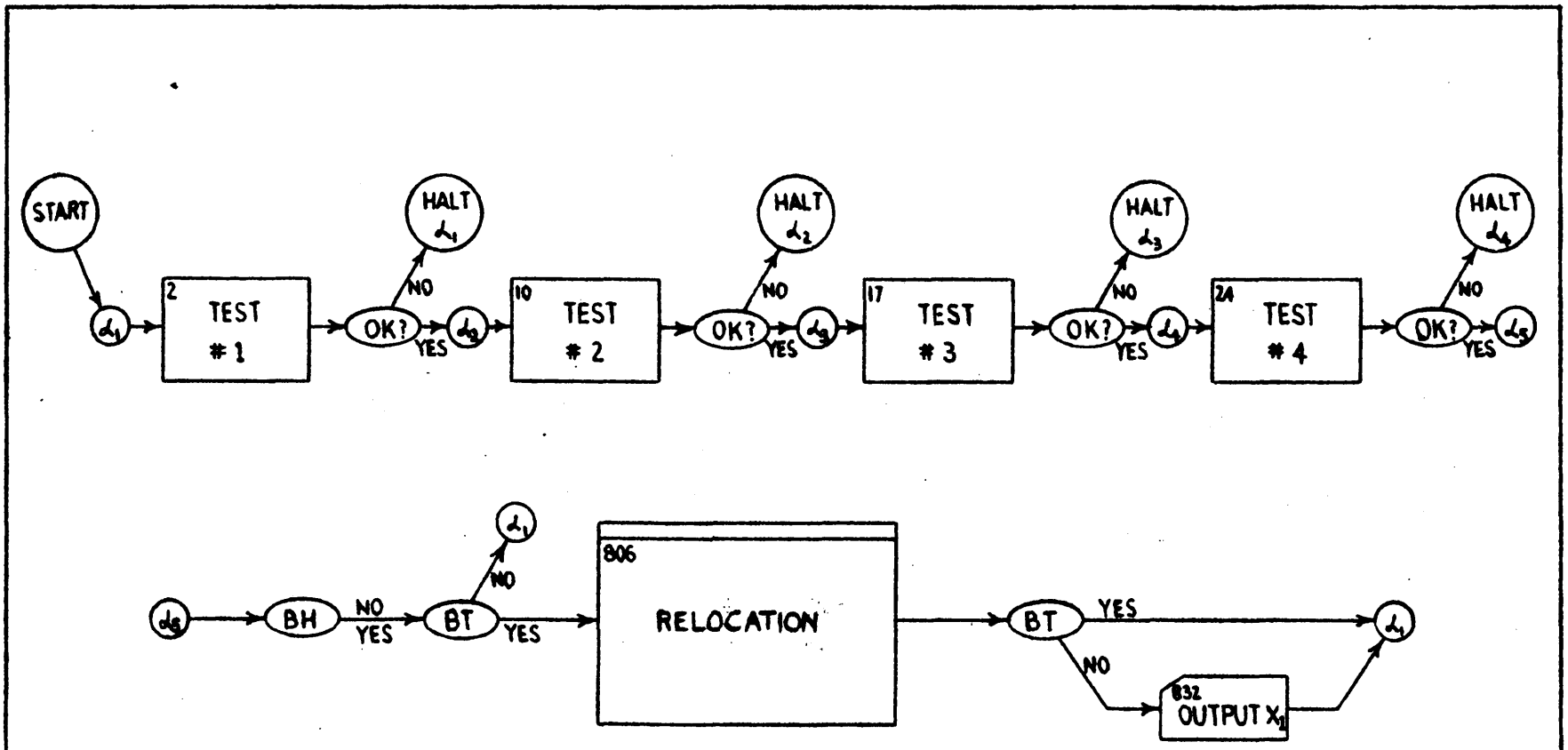


FIGURE 21
FLOW DIAGRAM OF THE
ACCEPTANCE TEST ROUTINE

routine pages 81 to 84 shows that the tests involved are for all practical purposes identical. Since the major difference between these two groups of tests is in the storage positions which they occupy, the test portion of the Acceptance Test Routine, hereafter referred to as the "test", will not be discussed any further except to note that in itself it is not a complete test of the machine since it does not test all tracks or minor memory positions.

(2) Relocation

The breakpoint transfer order at the end of the routine affords two modes of operation: If the BT switch is set to "No" the tests will be repeated, while if it is set to "Yes" the relocation portion of the routine will be entered. This part of the routine operates upon the test to move it to higher numbered storage locations and to adapt it to operate from its new locations.

When relocating the test this portion of the routine examines each instruction in the test starting with that in the highest numbered storage location and working toward the lowest. The routine, which is entered at order 806, extracts word W_i located in address x_i in the test into the opsto (845) reserved for this purpose. The word W_i is to be moved to a new address $x_i+\Delta$ where Δ is equal to the desired increase in storage location. There are $j=42$ words in the test of which $k+1 = 31$ are orders; the first order is located in storage location x_1. The last $j-(k+1)$ words are constants and are transcribed into their new locations $(x_i+\Delta)$ without being altered, while the first $k+1$ words, being orders, must be examined to determine if their A, B, and C addresses (α, β, γ) are

WISCoding for ACCEPTANCE TEST ROUTINE

By A. K. Scidmore Date _____ Page 1 of 4

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	0		I	[804]	[855]	[1023]		000	0	324	357	3ff
1.00	804		SA	x_i [848] -001	[1021]	[848]		000	b	350	3fd	350
1.01	805		SA	$(x_i+\Delta)$[849] -001	[1021]	[849]		000	b	351	3fd	351
1.02	806	1	E	x_i [848]	[2512]	[808]	IN	001	1	350	19c	328
1.03	807		A	x_1 [850] +k	[854]	[855]		000	8	352	356	357
1.04	808	1	E	W_i [X_1]	[150] = W_i	[845]		001	1	000	c12	34d
1.05	809		TN	(x_1+k)[855] -x_i	[848]	[827]	CONSTANT?	000	e	357	350	33b
1.06	810	37	E	W_i [845]	[1,4]	[846]		025	1	34d	014	34e
1.07	811		TZA	" [849] -001	[1019]	[817]	EXTRACT?	000	d	34e	3fa	331
1.08	812	13	E	W_i [845]	[1,12] = β	[846]		00d	1	34d	01c	34e
1.09	813		TN	β [849] -x_1	[850]	[817]	[B]	000	e	34e	352	331
1.10	814		TNA	j [853] -$(\beta-x_1)$[ ]		[817]	O.K.?	000	f	355	800	331
1.11	815		A	β [849] +Δ	[847] = $\beta+\Delta$	[ ]		000	8	34e	34f	800
1.12	816	1	E	" [ ]	[13,12]	[845]		001	1	800	0dc	34d
1.13	817	1	E	W_i [845]	[1,12] = γ	[846]		001	1	34d	01c	34e
1.14	818		TN	γ [849] -x_1	[850]	[822]	[C]	000	e	34e	352	336
1.15	819		TNA	j [853] -$(\gamma-x_1)$[ ]		[822]	O.K.?	000	f	355	800	336
1.16	820		A	γ [849] +Δ	[847] = $\gamma+\Delta$	[ ]		000	8	34e	34f	800
1.17	821	1	E	" [ ]	[1,12]	[845]		001	1	800	01c	34d
1.18	822	25	E	W_i [845]	[1,12] = α	[846]		019	1	34d	01c	34e
1.19	823		TN	α [849] -x_1	[850]	[123]	[A]	000	e	34e	352	07b
1.20	824		TNA	j [853] -$(\alpha-x_1)$[ ]		[123]	O.K.?	000	f	355	800	07b
1.21	825		A	α [849] +Δ	[847] = $\alpha+\Delta$	[ ]		000	8	34e	34f	800
1.22	826	1	E	" [ ]	[25,12]	[845]		001	1	800	19c	34d
1.23	827	1	E	$x_i+\Delta$ [849]	[1,12]	[829]		001	1	351	01c	33d

WISCoding for _____

By _____ Date _____ Page 2 of 4

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
1.24	828		A	[1023]	[1023]	[\ ]	IGNORE	000	8	3ff	3ff	800
1.25	829	1	E	W_i [845]	[1,50]	[$X_{1+\Delta}$]		001	1	34d	e12	000
1.26	830		TN	χ_1 [850] - χ_i [848]		[801]	ANY MORE?	000	e	352	350	324
1.27	831		BT	[—]	[—]	[834]		100	5	000	000	342
1.28	832		O	[848]	[848]	[834]		000	7	350	350	342
1.29	833		A	[—]	[—]	[\ ]	IGNORE	000	8	000	000	800
2.00	834	1	E	$\chi_i+\Delta$ [849]	[1,12]	[840]		001	1	351	01c	348
2.01	835		A	$\chi_i+\Delta$ [849] + j	[1023]	[848]	RESET χ_i	000	8	351	3ff	350
2.02	836		A	$\chi_i+\Delta$ [849] +0	[853]	[850]	RESET χ_1	000	8	351	355	352
2.03	837		A	χ_i [848] +Δ	[847]	[849]	RESET $\chi_i+\Delta$	000	8	350	349	351
2.04	838		TN	χ_j' [851] — "	[\]	[841]	TOO LARGE?	000	e	353	800	349
2.05	839		A	Δ_0 [855] +0	[1023]	[847]	RESET Δ	000	8	357	3ff	34f
2.06	840		TU	[—]	[—]	[X_1]	OUT	000	5	000	000	000
2.07	841		S	χ_j' [852] - Δ_0 [855]		[\]		000	a	354	357	800
2.08	842		S	" [\] - j [853]		[\]		000	a	800	355	800
2.09	843		S	χ_1' [851] — "	[\]	[847]	CHANGE Δ	000	a	353	800	34f
2.10	844		TU	[—]	[—]	[837]		000	5	000	000	345
	845			OPSTO FOR W_1	[ ]	[ ]		000	0	000	000	000
	846			OPSTO FOR $\alpha, \beta, [\gamma]$		[ ]		000	0	000	000	000
	847			OPSTO FOR Δ	[ ]	[15]		000	0	000	000	00f
	848			OPSTO FOR X_i	[ ]	[44]	INITIAL VALUES	000	0	000	000	02c
	849			OPSTO FOR $X_i+\Delta$	[ ]	[59]		000	0	000	000	03b
	850			OPSTO FOR X_1	[ ]	[2]		000	0	000	000	002
	851			X_1' [ ]	[ ]	[2]		000	0	000	000	002
	852			X_j' [ ]	[ ]	[801]		000	0	000	000	2e8

WISCoding for _____

By _____ Date _____ Page 3 of 4

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	853			j [ ]	[ ]	[42]		000	0	000	000	02a
	854			k [ ]	[ ]	[30]		000	0	000	000	01e
	855			Δ_0 [ ]	[ ]	[15]		000	0	000	000	011
			I	[1]	[44]	[2]		000	0	001	02c	002
	1			OPSTO]	[ ]	[ ]						
	2		A	O [1023] + O	[1023] = O	[46]		000	8	3ff	3ff	02e
	3		AA	$+\frac{c}{2}$ [34] + $(\frac{-3c}{2})$	[35] = 2c	[1]		000	9	022	023	001
	4		A	-3c [36] + "	[\ ] = -C	[45]		000	8	024	800	02d
	5		SPARE	[35]	[35]	[1]	IGNORE	000	4	023	023	001
	6		A	2c [1] + (-3c)	[36] = -C	[\ ]		000	8	001	024	800
	7	1	E	[\ ]	[1,50]	[46]		001	1	800	e12	02e
	8		TZ	-C [45] - C	[46]	[10]		000	c	02d	02e	00a
	9		H	[/]	[/]	[2]		000	6	000	000	002
	10		A	O [1023] + O	[1023] = O	[1]		000	8	3ff	3ff	001
	11	1	E	+K [37]	[1,47] = +8K	[1]		001	1	025	84f	001
	12		A	-K [38] + (-K)	[38] = -2K	[45]		000	8	026	026	02d
	13		S	5K [39] - 8K	[1] = -3K	[46]		000	a	027	001	02e
	14		S	-5K [40] - (-2K)	[45] = -3K	[\ ]		000	a	028	02d	800
	15		TZ	-3K [46] - (-3K)	[\ ] = O	[17]		000	c	02e	800	011
	16		H	[/]	[/]	[10]		000	6	000	000	00a
	17		A	O [1023] + O	[1023] = O	[45]		000	8	3ff	3ff	02d
	18		M	C_1 [41] x C_2	[42] = C_1C_2	[1]		000	2	029	02a	001
	19		M	C_2 [42] x C_3	[43] = C_2C_3	[\ ]		000	2	02a	02b	800
	20		D	" [\ ] ÷ C_1C_2	[1] = C_3/C_1	[\ ]		000	3	800	001	800
	21	4	E	" [\ ]	[1,47] = $C_3/8C_1$	[45]		004	1	800	81f	02d

WISCoding for _____

By _____ Date _____ Page 4 of 4

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	22		TZ	C_4 [44] − "	[45] = 0	[24]		000	e	02e	02d	018
	23		H	[—]	[—]	[17]		000	b	000	000	011
	24		TN	K [37] − 5K	[39]	[26]		000	e	025	027	01a
	25		H.	[—]	[—]	[24]		000	6	000	000	018
	26		TN	−5K [40] − (−K)	[38]	[28]		000	e	028	026	01c
	27		H	[—]	[—]	[26]		000	6	000	000	01a
	28		TN	−5K [40] − (−5K)	[40]	[27]		000	e	028	028	01b
	29		TN	K [37] − (−5K)	[40]	[27]		000	e	025	028	01b
	30		TZ	K [37] − (−5K)	[40]	[27]		000	e	025	028	01b
	31		BH	[—]	[—]	[32]		100	6	000	000	020
	32		BT	[—]	[—]	[806]		100	5	000	000	326
	33		T	[—]	[—]	[2]		000	5	000	000	002
	34			$+C/2$ [ ]	[ ]	[ ]						
	35			$-3C/2$ [ ]	[ ]	[ ]						
	36			$-3C$ [ ]	[ ]	[ ]						
	37			$+K$ [ ]	[ ]	[ ]						
	38			$-K$ [ ]	[ ]	[ ]						
	39			$+5K$ [ ] CONSTANTS		[ ]						
	40			$-5K$ [ ]	[ ]	[ ]						
	41			C_1 [ ]	[ ]	[ ]						
	42			C_2 [ ]	[ ]	[ ]						
	43			C_3 [ ]	[ ]	[ ]						
	44			$C_3/8C_1$ [ ]	[ ]	[ ]						
	45			[ ]	[ ]	[ ]						
	46			OPSTO [ ]	[ ]	[ ]						

"essential"[4], i.e., fall within the block of words which constitutes the present location of the test portion. If so, these addresses will have to be increased by the amount Δ unless, of course, the order is an extract order in which case bits 13 through 24 (normally the B address) must be left unchanged. When the last word has been relocated, the constants x_1, x_i, and $x_i+\Delta$ are reset to correspond to the new locations after which a check is made to determine that the new $x_i+\Delta$ will not exceed the last available storage location, x'_n. If it does, the value of Δ is changed from its usual positive value, Δ_o, to a negative value sufficiently large that the next relocation will place the test portion back near the first available storage x'_1.

Although a value of $\Delta_o=1$ could be used to move the test around the drum, the test would move so slowly that it would require something like 50,000 machine cycles (about 14 minutes) to move it forward 32 minor memory positions (1 track). To speed up the movement of the test a value of $\Delta_o=15$ is used, allowing the test to be moved forward by one track in approximately one minute. The storage locations occupied by the test will eventually be repeated, but before this occurs approximately 860,000 machine cycles or four hours will have passed. When the machine successfully performs all of these operations without detecting a single error the acceptance test will have been passed.

For preventive maintenance purposes, however, the time required by this routine is so great that a still larger value of Δ,

---

4. C. H. Davidson, "Mathematical Procedures and Testing Methods for an Intermediate Speed Digital Computer", Ph.D. Thesis, University of Wisconsin, 1952, 73.

perhaps $\Delta_o = 31$, must be used to expedite the testing of the various tracks. For convenience in identifying the exact position of the test an output order which can be reached by means of a BT is included in the relocation portion to punch out the constant x_1 each time the adaptation has been finished.

3. IntraComputational Test Routine

a. Philosophy and General Approach

The IntraComputational Test Routine, ICTR, is to be programmed during a problem, its purpose being to provide some degree of assurance that the machine is working correctly on the problem. Each successful passing of this routine will be interpreted as a reasonable guarantee of the reliability of the machine results up to that point.

In order to place absolute reliance upon the ICTR two assumptions would have to be valid: First, if the computer passes the ICTR at the beginning and end of some period of time it would also have passed it at any time between; and second, if the computer passes the ICTR, then it will also perform any other calculation correctly. The validity of the first assumption depends both upon how often the ICTR is programmed and upon the consistency of behaviour of the computer. With a reasonable degree of reliability, there should be no need to program the ICTR more often than every 3000 or so operations. Although this is an approximate recommendation as to frequency of use, the actual scheduling of the ICTR is greatly dependent upon the problem being performed and hence upon the judgment of the programmer. The most logical place to use it is after a major break in the computation. For example, in a long

program calling for a series of values to be calculated, the ICTR could be used to best advantage after each new value had been calculated. If the computer began to misbehave at some time during the problem, the results obtained up until the last successful passage of the ICTR might be saved and the calculation later resumed at that point, the validity of those results depending upon the type of failure found - intermittent, marginal, or catastrophic.

To make the second assumption valid would require that the ICTR be sufficiently comprehensive to test every function that could possibly be exercised by any program. For a discussion of this topic see section III A 1 a. In conflict to this is the requirement that the routine be brief so that it does not materially reduce the effective computing time or discourage the programmer from using it. Since some compromise must be reached, the routine proposed for use in the WISC does not test every possible combination of signs, numbers, etc. nor does it directly involve the input or output sections of the computer; but it does test a very large number of the machine operations -- and it is only 9 orders long.

b. The Routine

Since this routine involves so few orders, and the flow of calculation is so simple, no flow diagram of the routine will be presented; however, the routine in coded form may be found on page 88. Because of the importance of this routine; the functions of the orders in it will be discussed in detail below so that an analysis of its weak and strong points may be made.

Order 1.01 (E) Although the primary purpose of this order is not to test the machine, a machine error in its performance will

WISCoding for INTRA COMPUTATIONAL TEST ROUTINE

By A. K. Seidmore Date _____ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
1.01	990	13	E	[998]	[13,38]	[863]	3de	00d	1	3e6	8d6	35f
1.02	991		E	C [999]	[21,30]	[862]	3df	001	1	3e7	55e	35e
1.03	992		M	1/2 [1016] -C	[999] -C/2	[\]	3e0	000	2	3f8	3e7	800
1.04	993	20	E	C/2 [\]	[1,20]	[862]	3e1	014	1	800	414	35e
1.05	994		D	-C [862] ÷ 1/2	[1016]	[\]	3e2	000	3	35e	3f8	800
1.06	995		TN	-C [862] -(-2C)	[\] +C	[997]	3e3	000	e	35e	800	3e5
1.07	996		TZ	+C [\] -C	[999] =0	[998]	3e4	000	d	800	3e7	3e6
1.08	997		H	[/]	[/]	[997]	3e5	000	6	000	000	3e5
1.09	998		TU	[/]	[/]	[863]	3e6	000	5	000	000	35f
2.01	999			C The Test Constant		[ ]	3e7	000	5	6ef	e56	efe
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				Linkage Used To Call The ICTR								
α		13	E	[α+1]	1,12	[863]	L	00d	1	LH	01c	35f
α+1			TU	[~~β~~]	[~~β~~]	[990]		000	5	000	~~000~~ β	3de
				[Error May 20/58]		[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

probably be detected since the linkage from the ICTR back to the program will not be performed correctly. Its function is to extract bits 13 through 50 of the T order (1.09) into storage location 863 which is the operational storage (OPSTO) used for transportation back to the program, while bits 1 to 12 of the word in 863 are to be supplied by the program linkage orders.

Order 1.02 (E) This order extracts the first 30 bits of the constant C [999] into the opsto 862 shifting them twenty places to the left. Here, then, is an extract where $x<y$ and an even number of shifts are required. It should be noted that the constant C is "symmetrical" in nature since the first 20 significant bits, or 5 hexadecimal characters, are identical with the last twenty. Because the left-hand 10 bits are lost as a result of the shift, the contents of opsto 862 after this order has been completed will be 2fe.5 6ef e-- --- (a negative number).

Order 1.02 (M) Next the constant C is multiplied by 1/2 in effect shifting it one place to the right. The order calls for two positive operands and requires a short memory result.

Order 1.04 (E) The result of the preceding order is now shifted 19 places to the right with the resulting 20 low order bits being written into 862 by this order--an extract with $x>y$, short memory of the A operand, and an odd number of shifts. After order 1.04 has been completed the contents of opsto 862 (in hexadecimal form) should be 2fe.5 6ef e56 efe , which is treated by the fixed point version of thw WISC as -C since the exponent digits are discarded in the arithmetic unit.

Order 1.05 (D) Here the constant -C obtained as a result of 1.04 is divided by 1/2 with the result being -2C . This a division with A negative and B positive.

Order 1.06 (TN) The first test of the preceding operations is made by this TN order which subtracts -2C from -C and tests the sign of the result. Since the result should be +C the order should not cause a transfer. This is not a test on the exactness of the two operands; it only tests their relative magnitude and signs and, of course, some of the compare circuitry.

Order 1.07 (TZ) This order does effectively test the exactness of the results of the first 5 operations. Usually the result of the subtraction involved in a comparison order is not preserved; however, a recent minor modification of the result recirculator switch allows the result of any compare order to be "trapped" in the result recirculator even though writing of this result is blocked. Consequently, the result of a compare order is now available from "short memory". This order 1.07 calls for the result of the preceding order which is in effect a subtract order and compares this result, +C , with the original constant, +C . Now, if any errors have occurred in the preceding operations, the two operands will not be equal and the halt order, 1.08, will be reached.

Order 1.08 (H) The halt order may be reached if either of the two compare orders detect an error. Since the halt order transfers to itself, the machine will halt with this order in the order shift registers. This furnishes the operator with an indication that the ICTR has detected a machine malfunction.

Order 1.09 (TU) Should no error be detected by the preceding orders this order will transfer to the linkage order in 863.

Constant 2.01 Although other constants can be used, this constant was chosen for two reasons: First, there are two bit patterns which are read by the standard memory readers with the most difficulty--those involving a number of consecutive ones, and those with the sequence 10101--. Second, since it was desired that the number of orders and stored constants be kept to a minimum, a serial flow of calculations such that the final answer and the initial constant are equal would obviate the necessity of storing a separate constant to check the result. Since two extract orders are involved, and since an extract order involving any amount of shifting generally causes a loss of significant figures, it was decided to have these two extractions restore the original number. This is quite easily accomplished with a "symmetrical" constant such as this.

c. Accomplishments and Inadequacies

The orders in the ICTR can now be examined in the light of the discussion presented in section III A 1 a with reference to the factors affecting the operation of the WISC. As mentioned previously the orders presented on page 49 of this section were chosen with the preventive maintenance test routines in mind. Some of the combinations of factors were chosen rather arbitrarily with a minimum number of orders as the objective; there are undoubtedly other combinations of orders which will also adequately test the circuitry directly associated with each order. For this reason, the following separate analysis of the orders in the ICTR is presented.

Input and Output These orders are not involved in the ICTR since their inclusion would be so time consuming and such a nuisance to the operator that he would have good justification for not using the ICTR in his program.

Extract The extract orders involved test all of the circuitry directly associated with the extract circuits except the gate 7a of the illustrative example on page 46.

Multiply and Divide The multiply order programmed is for positive A and B operands while the divide order has one negative operand. Consequently, part of the result sign determining mechanism in the arithmetic Result Sign and Compare Output chassis is not completely checked by these two orders alone. Fortunately, the majority of this circuitry is also used for the add, subtract, and compare orders, and as a consequence these two orders in conjunction with the two compare orders do afford a complete test of the result sign generation as well as a test of the multiply and divide operations.

Add, Subtract, and Compare As mentioned in section III A 1 a above, these orders (A, AA, S, SA, TZ, TZA, TN, and TNA) involve much the same circuitry since there are only three operations which are performed on the significant digits of the operands. The compare orders call for two of these operations: subtract, and subtract and complement. Although, as mentioned above, the result sign determining circuits are adequately tested, the ability of the arithmetic unit to add the significant bits of the A and B operands correctly is not tested at all. This fortunately is not a major sin since the addition process does not differ greatly from the subtraction process and since the number of circuits that are not tested is very small.

Breakpoint and Unconditional Transfer Although the unconditional transfer is adequately tested since it is used to enter and leave the ICTR, the BT order is not tested at all. If the ICTR were programmed every 3600 machine cycles, testing the BT order would involve halting the machine and setting and resetting the Breakpoint Transfer switch every minute -- a process which would use up a considerable amount of available computing time and thoroughly tax the patience of the operator.

Halt and Breakpoint Halt Although the Halt order is included in the ICTR, it is not tested even if the ICTR detects an error since a failure to halt will go unnoticed if the operator is not expecting the machine to halt. Moreover, the testing of these two orders is out of the question for the reasons mentioned when discussing the BT order.

Other Factors The routine does test short memory coding of the A operand, B operand, and result, but tests only two of the possible read-write coincidences (Order and A Operand). Of course, all storage locations are not checked and although the block writing mechanism is tested, some of the orders (input, output, halt, and spare) which activate it are not included in the test routine.

Even though a test routine may be designed to exercise all of the circuits in the machine, certain of the more common machine failures may escape undetected unless certain precautions are observed. For example, suppose one test were to be made using the sequential orders D+E = F and F-E = D with the results of the second operation to be compared with the original constant, D , by means of a transfer on zero order. If now either the A or AA operand recirculators fail

to recirculate, the A operand presented to the arithmetic adder-subtractor chassis will be zero. The result of the two operations will then be 0+E = E and 0-E = -E if short memory coding (for the A operand) is not used in the second operation, and 0+E = E and E-E = 0 if it is. Then, a TZ order which calls for the result of this second order to be subtracted from the original constant (as the A operand) will detect or not detect the failure depending upon whether or not the second order uses a short memory coding for the A operand. Although the ICTR cannot be designed to detect every possible type of failure, the more common types of failures such as those caused by failure in the recirculators have been considered.

In summary, then, it can be considered that successful completion of the ICTR indicates that the following portions of the WISC are operating properly:

TIMING

Clock Pulse Generators
Clock Pulse Amplifiers and Power Pulsers
Timing Enable Generators
Arithmetic Digit Time Selectors
Minor Arithmetic Cycle Selector

CONTROL

Order Decoder -- Note 1
Order Shift Registers
R and S Shift Registers
Order Counter
Block Writing Control
A, B, Order, and A=B Serial Coincidence Detectors -- Note 2
Read-Write Time Coincidence Detectors -- Note 3
A Operand and Order Half Adders
Result Recirculator Switch
Extract and A Operand Recirculator Switch -- Note 4
Extract Circuits -- Note 4

MEMORY

Write Sequencer
Writing Time Locks
Writing Time Coincidence-detector
Writing Controls WDS, WTS, WFFL
Standard Memory Readers -- Note 5
Standard Memory Writer for Tracks 26 and 31
A, B, Order, and Result Recirculator Loops
Short Memory Control

ARITHMETIC

Arithmetic Order Type Decoder -- Note 6
Result Sign and Compare Output Generator -- Note 6
AA, BB, and RR Recirculator Loops
Precession Controls (Fixed Point Circuits)
Subtrahend Minuend Generator
Arithmetic Adder-subtractor
Arithmetic Control Generator -- Note 7

NOTE 1 Only the portions of the order decoder which are used for input, output, and halt orders are not tested.

NOTE 2 Although this serial concidence detector was originally intended to detect the completion of the input or output orders, it now is also used to determine when the extract shifting has been completed. Therefore, only gate 11b which is used on input and output orders only is not tested.

NOTE 3 Since a B operand-result read-write coincidence is not included in the ICTR, gate 3 and flip-flop 9 on this chassis are not tested.

NOTE 4 As mentioned previously gate 7a on the Extract and A Operand Recirculator Switch Chassis is not tested.

NOTE 5 The standard memory preamplifiers and track selectors for tracks other than tracks 26 and 31 are not tested.

NOTE 6 Since no "absolute" orders are called for, tube 3_a on the order decoder chassis and gate 4a and FF2 on the Result Sign and Compare output chassis are not adequately tested.

NOTE 7 Although only two of the eight add, subtract, and compare order types and only two of the four possible combinations of operand signs are used, almost all of the circuits in the arithmetic are tested by the ICTR. The only circuits not completely tested other than those mentioned in note 5 are gates 3b, 4a, 7a, and 8a, negator 9a, and flip-flop 1 on the Arithmetic Control Generator Chassis.

Not tested by the ICTR, then, are the following:

Input Unit and associated circuits
Output Unit and associated circuits
Halt order circuits
Breakpoint operation circuits
Console controls (slow speeds, "Stop", "Clear", etc.)
Reading and writing in all tracks and MM positions
The circuits mentioned above in notes 3, 4, 6, 7 (12 circuits in all)

It would seem, then, that this routine will be quite satisfactory even though it is only 9 orders long. Of the operations involving exclusively high speed operation over 99 percent of the circuits are tested by the ICTR.

Before concluding this discussion of the ICTR, it would be well to determine whether the routine could be improved so as to check some of the circuits presently not included. Because of the nature of the routine and its use, the testing of input, output, and halt orders, the breakpoint and console control circuitry, and all tracks and MM positions is out of the question. Any changes, then, would have to affect the 12 circuits mentioned above. To test all of these would require that the ICTR would have to be enlarged to include four more orders. While the inclusion of these four orders would not greatly affect the ratio of computing time to testing time, nor greatly reduce the number of storage spaces available for computation, the author feels, nevertheless, that enlarging the ICTR at this time would be inadvisable since to do so would involve a reallocation of the permanent storage locations (PERSTO). Such a reallocation would involve considerable amount of work since most of the subroutines now stored in PERSTO would have to be adapted to fit new locations. However, when the floating point unit is finished and put into use

the ICTR as well as the other routines stored in PERSTO will have to be modified to be compatible with the WISC in floating point form, and the possibility of such enlargement should be given serious consideration at that time.

B. Diagnostic Routines

1. General Philosophy

a. Why and When

Unavoidably, in a machine involving at least 1400 vacuum tubes, more than 10,000 resistors, and thousands of other components, failures will occur all too frequently. The speed and facility with which these faults are found and corrected will have a significant bearing on the useful amount of machine time available. Consequently, some provisions must be made for rapidly locating such faults so that the computer can be restored to reliable operation with as little loss of computing time as is possible. As mentioned in the introduction to this thesis, the approaches to this problem are of two types: those involving human observation of waveforms, indicator lights, and answers from the machine, and those which involve the decision making ability of the machine to help localize its own fault. The routines in this section are based upon the second approach as far as is possible because of the rapidity with which a malfunction can be located.

The test routines presented so far ask only one general question: is the machine working correctly? When the existence of a malfunction has been discovered by any test routine, or for that matter by any other means, the next obvious question is: why did it occur?, or more specifically: where is the cause of the malfunction to be found? what must be done to restore normal operation? If the failure was induced during preventive maintenance by the insertion of a marginal checking voltage, the error may be already localized to some extent. If, however, the

error occurred (for example) during the performance of the ICTR the only information available is that the error did occur unless, of course, the reason for the error were a power supply failure or other obvious cause. In such a situation, with a machine error detected, but no information concerning its cause readily available, the diagnostic routines will be called in to play to shed as much light as is possible on the specific cause of the disorder.

b. Possibilities and Limitations

The ideal diagnostic test routine would be one which would require no human attention other than that required for loading the routine and reading the exact location of the faulty component from the results printed out. Unfortunately this specific a diagnosis of the trouble often is neither practical nor feasible. It is impossible, for example, to determine by means of a program which circuit in a recirculator loop is adding bits to the information being recirculated. Likewise, since a malfunction occurring in many of the clock, timing, or control circuits may make it impossible even to load the machine, a diagnostic routine cannot be used to supply information identifying the specific cause. Therefore diagnostic routines are not only limited in application to those circuits the operation of which can be varied by means of a program, but even here the degree to which these routines can localize the failure varies widely from circuit to circuit.

c. Assumptions

(1) Logical Design

In order to design diagnostic test routines for a computer, it is necessary to examine in detail not only its overall logical

design but that of its subsections as well, for only from such a study can the questions to be asked of the machine by the diagnostic test routines and the sequence in which they are to be asked be intelligently determined. Since the design of these test routines is so dependent upon the logical design of the computer, it is essential that the actual physical construction of the machine conform exactly to the logical design as set down on paper or the results of the diagnostic test routines will be misleading or senseless. The possibility that a computer, especially the WISC, might not conform exactly to the drawings which purport to represent it is not as remote as it might at first seem. Because of the lack of continuity of personnel and the many revisions made during the evolution of the WISC many such discrepancies did exist. However, all such differences between the "paper" design and the actual machine should have been corrected during the debugging and testing of the computer; hereafter, it will be assumed that for the purposes of designing and using the diagnostic routines the machine corresponds exactly to its design drawings.

(2) Physical Structure

As mentioned above, the use of diagnostic routines is limited to those circuits the operation of which can be varied by means of a program. Hence, a fault in such physical portions of the machine as the power supplies, drum tape units, connections, etc. cannot in general be diagnosed by any routine; instead, such a fault must be repaired before the diagnostic routines can be used to locate any malfunctions. Consequently, the use of these routines is based on the assumption that these physical parts of the machine are sound.

(3) MINIWOC

In order to obtain any intelligent response from the computer a certain minimum of circuitry must be in operating condition. Since the starting point in the design and use of the Diagnostic Test Routines, then, begins with this minimum working computer, or MINIWOC, it should be defined in fairly explicit terms.

For the purpose of this discussion the MINIWOC will be assumed to have the ability to perform correctly the input, output, and halt orders. Although the machine could accept instructions and output answers without being able to stop itself, it must be possible at least to start and stop the machine manually, and since the additional circuits required to decode and perform a halt order are so few the convenience afforded by its presence in the MINIWOC easily justifies its inclusion. The portions of the WISC which must be functioning in order to perform the input, output, and halt orders correctly are listed below:

TIMING - all except arithmetic timing sections

CONTROL

Order Decoder
All Shift Registers
Order Counter
A, Order, and A=B Serial Coincidence Detectors
A and Order Half-Adders
Block Writing Control
Order Read-write Coincidence Detector
Result Recirculator Switch
Extract and A Operand Recirculator Switch

MEMORY

Input and Write Sequencer
Writing Time Locks
Write Time Coincidence Detector
Writing Controls WDS, WTS, WFFL
Standard Memory Readers
Standard Memory Writers
A, Order, and Result Recirculator Loops

INPUT - ALL

OUTPUT - ALL

A few of the circuits in the above sections are not used for the input, output, or halt orders, and others may or may not be included depending upon the program used. For example, since reading and writing is restricted to tracks zero and one during the initial diagnostic routines, the circuits associated with reading and writing in the other tracks are not included in the MINIWOC.

From the above list it can be seen that a large percentage of the machine must be working properly before any diagnostic test routines can be used. This is one of the main limitations in the use of diagnostic test routines for the WISC. Fortunately, the diagnosis of faults in much of the MINIWOC is not difficult since the location of these faults is often obvious from the symptoms displayed.

(4) Single Faults

A fundamental assumption upon which the diagnostic routines are based is that the malfunction to be diagnosed is due to a single fault. Although more than one fault may actually exist at a particular time, this should be the exception rather than the rule if preventive maintenance procedures are followed faithfully. If a number of faults are allowed to accumulate in the machine before any attempt is made to locate and repair them, their diagnosis by any means may become a very difficult task.

d. Sequence of Testing

Before presenting further discussion of the parts of the machine to be tested by these diagnostic routines, the sections not included in MINIWOC should be definitely determined; these are:

TIMING - arithmetic section

CONTROL

Order Decoder (other than Input, Output, and Halt)
Block Writing Control (for short memory of result and compare order)
B Operand Serial Coincidence Detector
Result Recirculator Switch (Arithmetic and Extract)
Extract and A Operand Recirculator Switch (Extract)
Extract Circuits

MEMORY

Standard Memory Readers
Standard Memory Writers
B Recirculator Loop
Short Memory Control

ARITHMETIC - all

Assuming that no information concerning the location of the fault is available, none of the circuits can be considered to be above suspicion. Because the circuits associated with the transfer order and breakpoint operations involve very little circuitry above that required for the halt order, they will be tested first. Since it is desirable to have the machine make most of the decisions concerning its behavior, the next routines should establish whether or not the fault exists in the compare circuits. Although these first tests are applied to a minimum of circuitry, a failure here may do no more than suggest areas where the cause may be found. However, once the ability of the machine to compare numbers--even if only in relative magnitude and sign--has been established a more specific diagnosis becomes possible. With the compare circuits working, it is a fairly easy task to test the operation of the remainder of the arithmetic. The order in which the remaining portions of the machine are tested is not of any particular consequence as long as each new

circuit or group of circuits is tested using only those part of the machine proven to be working properly.

e. Procedure Following the Discovery of an Error

What are the possible procedures which may be followed when one of the orders in a diagnostic routine indicates that a failure has occurred during the performance of that routine?

(1) Continue the remainder of the routine

Since the performance of each order may involve a large number of circuits, a number of which may not have been definitely proven to be operating correctly, a single error may not provide a great deal of information concerning the location of the fault. The fault may affect a number of the operations in a routine, and from the results of all of these operations a more specific diagnosis usually is possible. Consequently, it will be the general policy in these diagnostic test routines to continue the routine to obtain as much information as is possible before halting or punching out answers.

(2) Print Results

The results of a routine should of course be communicated to the operator. In case an error (or errors) has been discovered during the performance of one of these routines, the computer should then punch out answers indicating what has happened and what conclusions (if any) it has drawn concerning the location of the malfunction. In some cases the output may merely indicate what orders have been incorrectly performed while in others it may examine the pattern of failures and provide some indication of the location of the fault. In the former case the operator

will have to examine the failure pattern and consult a table showing the failures to be expected from various faults.

(3) Halt

When the results have been communicated to the operator the machine may be called upon to halt while the operator decides upon the procedure to be followed next.

(4) Reiterate

In the case of intermittent faults, it may be necessary to repeat a particular routine a number of times. Consequently, some provisions will be incorporated into the routine to provide for closed loop operation. For instance, in the location of intermittent shorts it may be desirable to have the machine perform the routine continuously at high speed while chassis are jarred or individual tubes are tapped in an attempt to induce failure often enough to locate the trouble.

(5) Load New Instructions or Operands

Since a particular diagnostic routine will rarely be able to specify the exact location of the fault, it will frequently be desirable to load new instructions or operands in an attempt to further delimit the fault. These new instructions may be in the form of another diagnostic routine or a special test routine to be used with test equipment such as an oscilloscope. New operands may be inserted to test the machine further with the same diagnostic routine but different constants or to provide an easily read oscilloscope pattern for further tracing.

f. Marginal Checking

The marginal checking system can be very useful when used in conjunction with diagnostic routines as well as when used with preventive

maintenance routines. For example, during preventive maintenance testing a deteriorated component may be discovered and by means of the marginal checking breakdown localized in a particular line group. It may still be difficult to locate this component from the small amount of information obtained through the use of the preventive maintenance routines. To aid in the location of the aforementioned component a diagnostic routine which tests the suspected circuits may be run while the marginal checking voltage is applied. In the above example the marginal checking system is used to duplicate the conditions under which the fault was detected.

Marginal checking voltages may also be used to aggravate a malfunction. Faults of an intermittent or marginal nature are the most difficult to diagnose even with the aid of diagnostic routines. By the use of a marginal checking voltage these marginal faults may be converted (at least temporarily) into consistently recurring or "dead" faults which are much easier to locate.

2. Testing the MINIWOC

Perhaps the fastest and simplest way to determine whether the MINIWOC is available is to test completely the input, output, and halt orders. Since the question to be answered is whether or not the computer is capable of processing these three orders correctly, a reliability rather than a diagnostic routine is indicated. There would seem to be no need, then, to design a completely new routine for testing the MINIWOC since the Input-Output Preventive Maintenance Test Routine, Section III A 1 b (1), is designed to test the input, and output orders thoroughly. The only complications arise from the inclusion of the breakpoint transfer order and the omission of any provisions to indicate which of the orders in the

routine (if any) fail to transfer to their C address. However, this routine can be adapted rather easily to fit this application, and a modification of it (see section III B 3 b) will be used to test for the presence of the MINIWOC.

What should be done next if the results of the aforementioned routine indicate that the MINIWOC is not present? A detailed description of the steps to be taken to diagnose the trouble in this case would be little more than a trouble shooting manual; accordingly, only the general points to be considered will be presented here.

Power Supply The power supply voltages and fuse indicators can be checked almost at a glance. It should not be necessary to monitor the currents drawn by each supply.

External Controls Correct operation of the "Start", "Stop", and "Clear" controls can be ascertained by observing the indicating lights on the console, while the slow speed operations can be checked by observation of the neon indicators on the one pulser or of the time taken per machine cycle.

Timing and Control Provisions are now being made to bring the majority of the timing signals, major control enables, and information paths, out to banana jacks on the front of the unit presently housing the control section of the WISC where these signals can be observed with an oscilloscope should they be suspected.

Input and Output Units In performing an input order the information is read from the tape, copied into the input buffer memory, trapped in the result recirculator, and then written in standard memory. During an output order the words in the standard memory are first trapped in the A Operand recirculator, then written in the output intermediate memory, and

finally read from there and punched on the paper tape. Should an error occur, the information may be traced as it goes along the paths indicated above. If the fault is in the control of the input or output units, some information can be gathered concerning the fault by using the buttons and switches on the console. For instance, the action of the "Clear Full", "Clear Empty", "Start", and "Stop" buttons on the input unit can be used to make a partial check of the operation of the counter-coincidence detectors, the Describer chassis, and the Tape-Reader chassis.

3. The Routines

a. The ICTR test - DTR1

Assume that an error has occurred during the performance of the ICTR. Should the location and character of this fault be obvious, it may be possible to repair it very quickly. Once the above faulty component has been replaced it may be tempting to assume that the machine is again in proper working order. However, before the program is resumed it would be wise to test the computer using the ICTR, to determine whether the original fault has been propagated to other parts of the machine or even whether the replacement has completely corrected the fault. Diagnostic Test Routine 1 (DTR1) presented on page 109 consists of the linkage required to set up a test loop using the ICTR. Although the ICTR is fundamentally a reliability routine, some diagnosis may be accomplished using it in the manner indicated above. Because of its simplicity, no flow diagram will be shown.

b. The MINIWOC Test - DTR2

As mentioned in section III B 2 some routine should be provided to test for the presence of the MINIWOC. The routine on page 110, Diagnostic Test Routine 2 (DTR2), is to be used for that purpose.

WISCoding for DIAGNOSTIC TEST ROUTINE 1 (DTR1)

By A. K. Scidmore Date ______ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
0.00	0		I	[0]	[1]	[0]		000	0	000	001	000
1.00	0	13	E	[001]	[1,12]	[863]		00d	1	001	01c	35f
1.01	1		T	[/]	[/]	[990]		000	5	000	000	3de
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

WISCoding for DIAGNOSTIC TEST ROUTINE 2 (DTR2)

By A. K. Scidmore Date _____ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
1.00	0		I	[1]	[40]	[2]		000	0	001	01f	002
1.01	1		0	[6]	[6]	[1000]	001	000	7	006	006	3e8
1.02	2		H	[/]	[/]	[4]	2	000	6	000	000	004
1.03	3		0	[7]	[7]	[1000]	3	000	7	007	007	3e8
1.04	4		0	[0]	[40]	[2]	4	000	7	000	01f	002
1.05	5		0	[8]	[8]	[1000]	5	000	7	008	008	3e8
	6			[ ]	[ ]	[ ]	6	123	4	b01	b01	b01
	7			[Keys]	[ ]	[ ]	7	123	4	b02	b02	b02
	8			[ ]	[ ]	[ ]	8	123	4	b03	b03	b03
	9			[ ]	[ ]	[ ]	9	001	0	fac	ade	001
	10			[ ]	[ ]	[ ]	a	002	0	fac	ade	002
	11			[32 Cpnstants	]	[ ]	b	003	0	fac	ade	003
	12			[ ]	[ ]	[ ]	c	004	0	fac	ade	004
	⋮			[ ]	[ ]	[ ]						
	⋮			[ ]	[ ]	[ ]						
	40			[ ]	[ ]	[ ]		032	0	fac	ade	032
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

Again no flow diagram is needed, but from an examination of the routine it can be seen that if the MINIWOC is present, the routine consists of only 3 orders: an input, an output, and a halt. In the event that any of these three orders fail to cause a transfer to its C address so that the next sequential order is read instead, the machine may decode the corresponding output order normally by-passed. The function of these three output orders is to cause the punching out of a key constant (hereafter referred to as a key) indicating the error that has occurred.

The selection of the test constants shown here as well as the points to be considered in testing the input and output units were discussed in some detail in section III A 1 b (1).

c. The Transfer-Breakpoint Test - DTR3

As mentioned above (section III B 1 d) the circuits associated with the transfer order and the breakpoint operations will be tested first. These operations are to be checked by Diagnostic Test Routine 3 (DTR3) which is shown in flow diagram form in Figure 22 and in coded form on page 113.

There are two minor loops in this routine. The first loop, used whenever both breakpoint switches are set to "No", includes only five orders in sequence (BT, BH, TU, H and BT). With these settings none of the four output orders (see Figure 22) should be reached. However, if an error occurs and an output order is decoded, a key constant will be punched out to indicate the location in the routine of the malfunction. The second loop, which may be reached by setting the BT switch to "Yes", consists of only three orders: a BT, a BH, and a H. Again, if the output order shown is decoded, another key

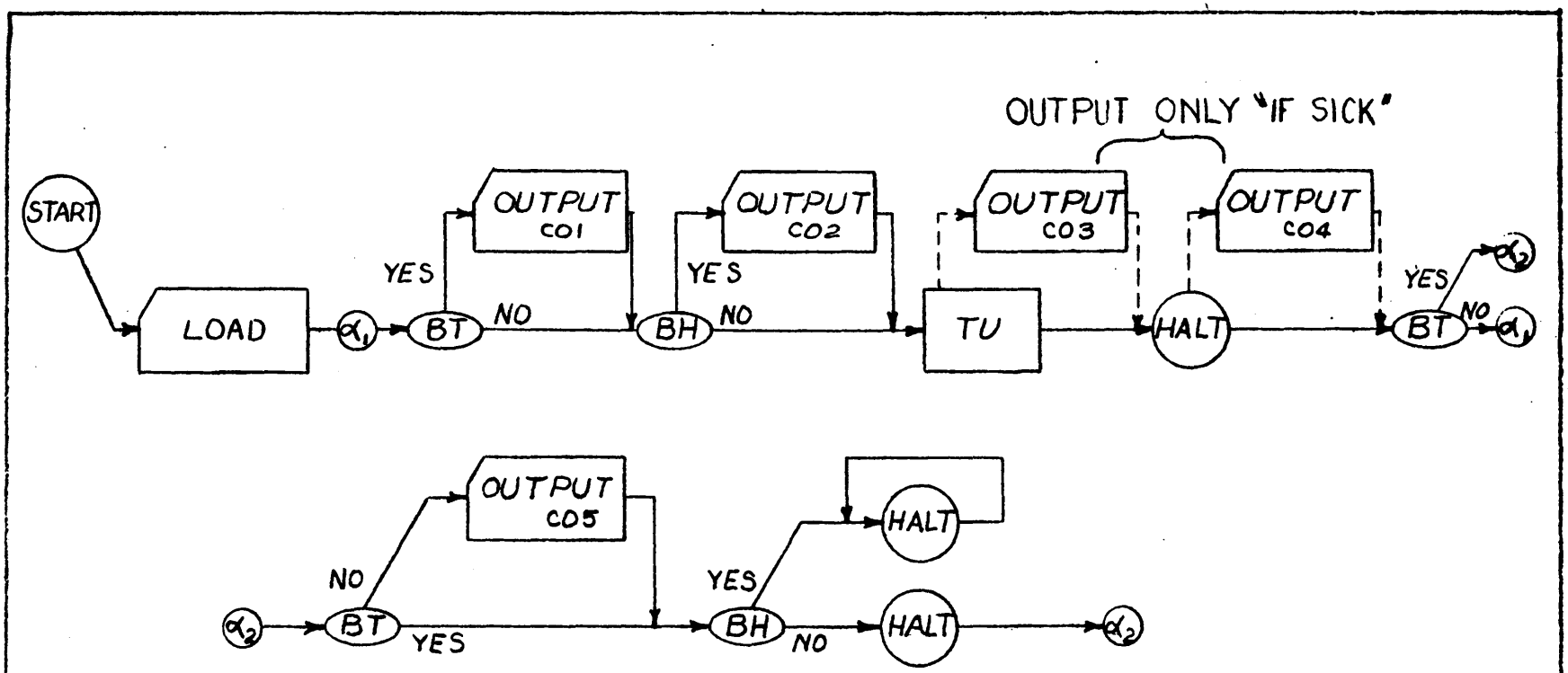


FIGURE 22
FLOW DIAGRAM OF DIAGNOSTIC
TEST ROUTINE 3 (DTR-3)

WISCoding for DIAGNOSTIC TEST ROUTINE 3 (DTR3)

By A. K. Scidmore Date _____ Page 1 of 1

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	0		I	[1]	[19]	[1]		000	0	001	013	001
1.10	1		BT	[/]	[/]	[5]	001	100	5	000	000	005
1.20	2		BH	[/]	[/]	[6]	2	100	6	000	000	006
1.30	3		TU	[/]	[/]	[7]	3	000	5	000	000	007
1.31	4		O	[17]	[17]	[7]	4	000	7	011	011	007
1.11	5		O	[15]	[15]	[2]	5	000	7	00f	00f	002
1.21	6		O	[16]	[16]	[3]	6	000	7	010	010	003
1.40	7		H	[/]	[/]	[9]	7	000	6	000	000	009
1.41	8		O	[18]	[18]	[9]	8	000	7	012	012	009
1.50	9		BT	[/]	[/]	[11]	9	100	5	000	000	00b
1.60	10		TU	[/]	[/]	[1]	a	000	5	000	000	001
2.10	11		BT	[/]	[/]	[13]	b	100	5	000	000	00d
2.11	12		O	[19]	[19]	[13]	c	000	7	013	013	00d
2.20	13		BH	[/]	[/]	[1000]	d	100	6	000	000	3e8
2.30	14		H	[/]	[/]	[11]	e	000	6	000	000	00b
	15			[ ]	[ ]	[ ]	f	123	4	c01	c01	c01
	16			[ ]	[ ]	[ ]	010	123	4	c02	c02	c02
	17			[Keys]	[ ]	[ ]	011	123	4	c03	c03	c03
	18			[ ]	[ ]	[ ]	12	123	4	c03	c04	c04
	19			[ ]	[ ]	[ ]	13	123	4	c05	c05	c05
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

constant will be punched out indicating that an error has been made. The last order, the halt order in OPSTO 1000, may be reached from this second loop by setting the breakpoint halt switch to "Yes". This last order actually forms a one word loop since it transfers to itself.

Obviously, the output orders included in this routine will not catch all of the possible errors which may occur. However, since the MINIWOC should be present, many possible errors should be eliminated, and only failure in those circuits not included in the MINIWOC need be considered. Nevertheless, this still leaves the possibility that the computer may fail in such a way that the output orders are never decoded. If this occurs the machine will leave the routine and diagnosis will have to be done using the slow speeds and manual operation in conjunction with visual observation of the indicator lights on the console.

The routine should be entered with both breakpoint switches set to "No". The first loop in the routine which provides for the testing of the BT ("No"), BH ("No"), TU, and H orders will then be set up. After the computer has performed the operations in this loop to the operator's satisfaction, BT ("Yes") operation may be tested by changing the setting of this switch, entering the second loop. Finally the operation of BH ("Yes") may be checked by changing the BH switch to the "Yes" position.

d. The Compare Test - DTR4

As mentioned previously, the diagnostic routines for the WISC will employ--as far as possible--the decision making ability of the machine to help locate its own faults. This routine, Diagnostic

Test Routine 4 (DTR4), is the first which actually uses this ability--in the form of the compare orders--to aid in diagnosing its own faults; DTR1 is actually a reliability routine, DTR2 is used to test for MINIWOC, while DTR3 employs actual machine failures, not its decision making features, to aid in the diagnosis of faults.

The flow diagram of this routine is shown in Figure 23, and the routine in coded form appears on pages 117 and 118. At the start of the routine the BT switch should be set to "Yes" so that the loop involving the compare test will be formed. Should the machine decode an output order while operating on the orders in this loop, a key constant will be punched out indicating which order failed to transfer (or transferred improperly). In this event, the results of the subtract operations will also be desired; to obtain these the BT switch should be set to "No" and, after the machine halts, reset to "Yes". Then when the machine is restarted these results will be punched out, and the machine will halt preparatory to entering the compare test again. If, however, the computer does not decode any output orders while operating on the compare test loop, the next diagnostic routine should be loaded. This can be accomplished by setting the BT switch to "No" and, after the machine has halted, merely restarting.

The main function of the routine is to test that part of the WISC associated exclusively with the compare orders: i.e., those circuits in the Order Decoder (OD) and Result Sign and Compare Output (RSCO) chassis which determine whether or not a transfer shall take place. Since the compare orders involve a subtraction of operand B from operand A, it is impossible to test the aforementioned circuits

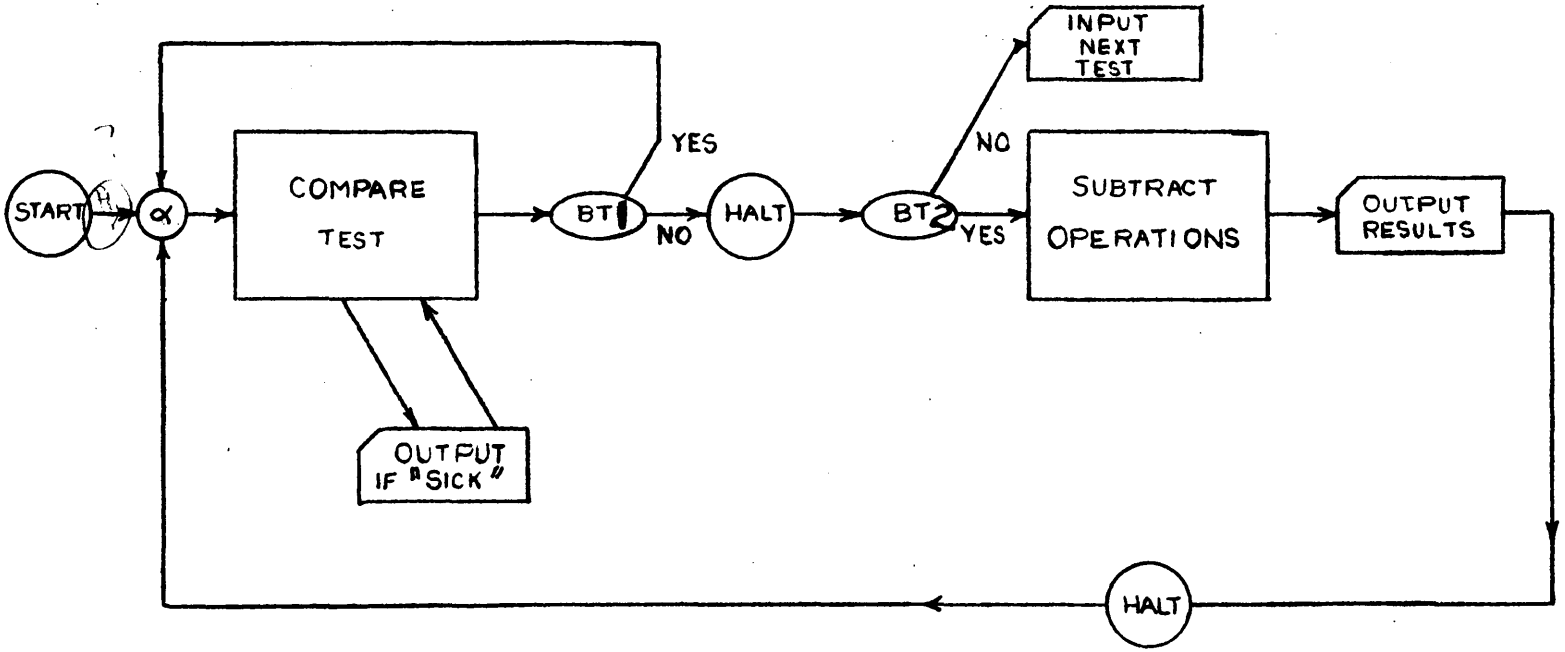


FIGURE 23
FLOW DIAGRAM OF DIAGNOSTIC TEST
ROUTINE 4 (DTR-4)

WISCoding for DIAGNOSTIC TEST ROUTINE 4 (DTR4)

By A. K. Scidmore Date _____ Page 1 of 2

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
0.00	0		I	[1]	[39]	[1]		000	0	001	027	001
1.00	1		TZ	[1023]	[1023]	[3]	001	007	e	3ff	3ff	003
1.01	2		0	[25]	[25]	[3]	2	007	7	019	019	003
1.10	3		TN	[1023]	[31]	[5]	3	008	e	3ff	01f	005
1.11	4		0	[26]	[26]	[5]	4	008	7	01a	01a	005
1.20	5		TN	[32]	[1023]	[7]	5	009	e	020	3ff	007
1.21	6		0	[27]	[27]	[7]	6	009	7	01b	01b	007
1.30	7		TN	[33]	[32]	[14]	7	010	e	021	020	00e
1.40	8		TN	[31]	[33]	[15]	008	011	e	01f	021	00f
1.50	9		TN	[33]	[1023]	[16]	9	011	e	021	3ff	010
1.60	10		BT	[/]	[/]	[1]	a	100	5	000	000	001
2.00	11		H	[/]	[/]	[12]	b	000	6	000	000	00c
2.10	12		BT	[/]	[/]	[17]	c	100	5	002	000	011
3.00	13		I	[/]	[/]	[/]		000	0	000	000	000
1.31	14		0	[28]	[28]	[8]		000	7	01c	01c	008
1.41	15		0	[29]	[29]	[9]		000	7	01d	01d	009
1.51	16		0	[30]	[30]	[10]	010	000	7	01e	01e	00a
2.20	17		S	[1023]	[1023]	[34]	1	007	a	3ff	3ff	022
2.30	18		S	[1023]	[31]	[35]	2	008	a	3ff	01f	023
2.40	19		S	[32]	[1023]	[36]	3	009	a	020	3ff	024
2.50	20		S	[33]	[32]	[37]	4	000	a	021	020	025
2.60	21		S	[31]	[33]	[38]	5	00b	a	01f	021	026
2.70	22		S	[33]	[1023]	[39]	6	00c	a	021	3ff	027
2.80	23		0	[34]	[39]	[24]	7	000	7	022	027	018
2.90	24		H	[/]	[/]	[1]	8	000	6	000	000	001

WISCoding for DIAGNOSTIC TEST ROUTINE 4 (DTR4)

By _____ Date _____ Page 2 of 2

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	25			[ ]	[ ]	[ ]	019	123	4	d01	d01	d01
	26			[ ]	[ ]	[ ]	01a	123	4	d02	d02	d02
	27			[ ]	[ ]	[ ]	01b	123	4	d03	d03	d03
	28			[Keys	[ ]	[ ]	01c	123	4	d04	d04	d04
	29			[ ]	[ ]	[ ]	d	123	4	d05	d05	d05
	30			[ ]	[ ]	[ ]	e	123	4	d06	d06	d06
	31			[ ]	[ ]	[ ]	f	000	0	fff	fff	fff
	32			[Constants	[ ]	[ ]	020	200	0	fff	fff	fff
	33			[ ]	[ ]	[ ]	1	200	0	000	000	000
	34			[ ]	[ ]	[ ]	2			0		
	35			[ ]	[ ]	[ ]	3			+C		
	36			[OPSTO For	Subtract	Operations	4			-C		
	37			[ ]	[ ]	[ ]	5			+C		
	38			[ ]	[ ]	[ ]	6			+C		
	39			[ ]	[ ]	[ ]	7			0		
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

without also testing a large number of other circuits. For example, the AA, B, BB, RR recirculator loops, the Arithmetic Adder-Subtractor, the Arithmetic Control Generator, and many other previously untested portions of the machine are all involved in some manner in these orders. Consequently, this test was made as elementary as possible in an attempt to minimize the effect of these other circuits on its operation. As can be seen from an examination of the coded routine and its constants, the operands and results should be either zero or a constant C which in binary form is almost all ones. With the exception of the TZ order the exactness of the result of the comparison is not relied upon--only the magnitude and sign. Should a recirculator loop be dropping bits (one of the most common faults found in these circuits), it would have to drop all of them before orders 1.10. 1.20, 1.30 or 1.40 would fail to transfer (or would transfer improperly), and such a fault would not affect the operation of orders 1.00 or 1.50 in the least. Nevertheless, an error detected by this test may be due to a multitude of causes.

Should the compare test be successfully passed, it is a certainty that those circuits which are exclusive to the compare orders are working properly, and they may subsequently be used to test other circuits. On the other hand, if the compare test does fail it cannot be said for certain that it is these compare circuits which are misbehaving and not the arithmetic circuits themselves. However, if the results of these subtractions are made known, it can be determined whether the trouble is in performing the subtraction or in making the decision. If the results of the subtractions are correct, the fault is in the compare circuits, and in this case it

is possible to determine the cause and approximate location of the fault from a comparison of the key outputs with the "cause and effect table" shown in Figure 24 which lists the faults which may be encountered in the compare circuits and the pattern of failures produced by each.

If the results of the subtractions themselves are not correct, however, the fault is not in the compare circuits, and they can be used to test other circuits. Furthermore, from an examination of the results of these subtractions it may be possible to ascertain approximately the nature of the arithmetic fault. Should the results give no such indication it will be necessary to proceed to the next sequential diagnostic routine.

e. The Add-Subtract Test (DTR5)

Following the successful passing of DTR4, the next logical group of circuits to be tested are those utilized in the add, subtract, and compare orders to generate a numerical answer from the A and B operands. These are the circuits which are to be checked by Diagnostic Test Routine (DTR5) which appears in coded form on pages 123-125 and in flow diagram form in Figure 25.

From a comparison of Figure 25 with Figure 23 it can be seen that the form of DTR5 is identical with that of DTR4. Hence, the procedure outlined in connection with the use of DTR4 is also applicable here. The smaller loop encompassing the add-subtract test is used to determine whether or not these circuits are functioning properly. If so, the next sequential diagnostic routine will be loaded, while if not the results of the add-subtract test will have to be punched out. Using the key constants and the results of the add-subtract test the location of the

FIGURE 24

CAUSE AND EFFECT TABLE FOR DTR4

Key	CAUSE													Subtraction Results
	1	2	3	4	5	6	7	8	9	10	11	12	13	
d01	X	X	X											+0
d02		X	X	X	X	X	X	X						-C
d03		X	X	X			X	X	X	X	X			-C
d04						X					X	X	X	+C
d05										X		X	X	+C
d06			X										X	-0

CAUSE:

1. CN = 1, RSCO chassis $G_{18A} = 0$, $N_{18B} = 0$
2. Never Transfers (T = 0)
3. RSCO chassis FF_{23}, $CF_{24B} = 1$
4. RSCO chassis FF_{23}, $CF_{24} = 0$, $G_{17A} = 0$
5. RSCO chassis $G_{10A} = 0$
6. RSCO chassis FF_{8}, $CF_{9} = 1$
7. RSCO chassis G_{13A}, FF_{14}, $CF_{15} = 0$
8. RSCO chassis G_{12B}, $FF_{11} = 0$, CN = 0
9. RSCC chassis $G_{10B} = 0$
10. RSCO chassis FF_{14}, $CF_{15} = 1$
11. RSCO chassis G_{7B}, FF_{8}, $CF_{9} = 0$
12. RSCO chassis $FF_{11} = 1$
13. Always Transfers (T = 1)

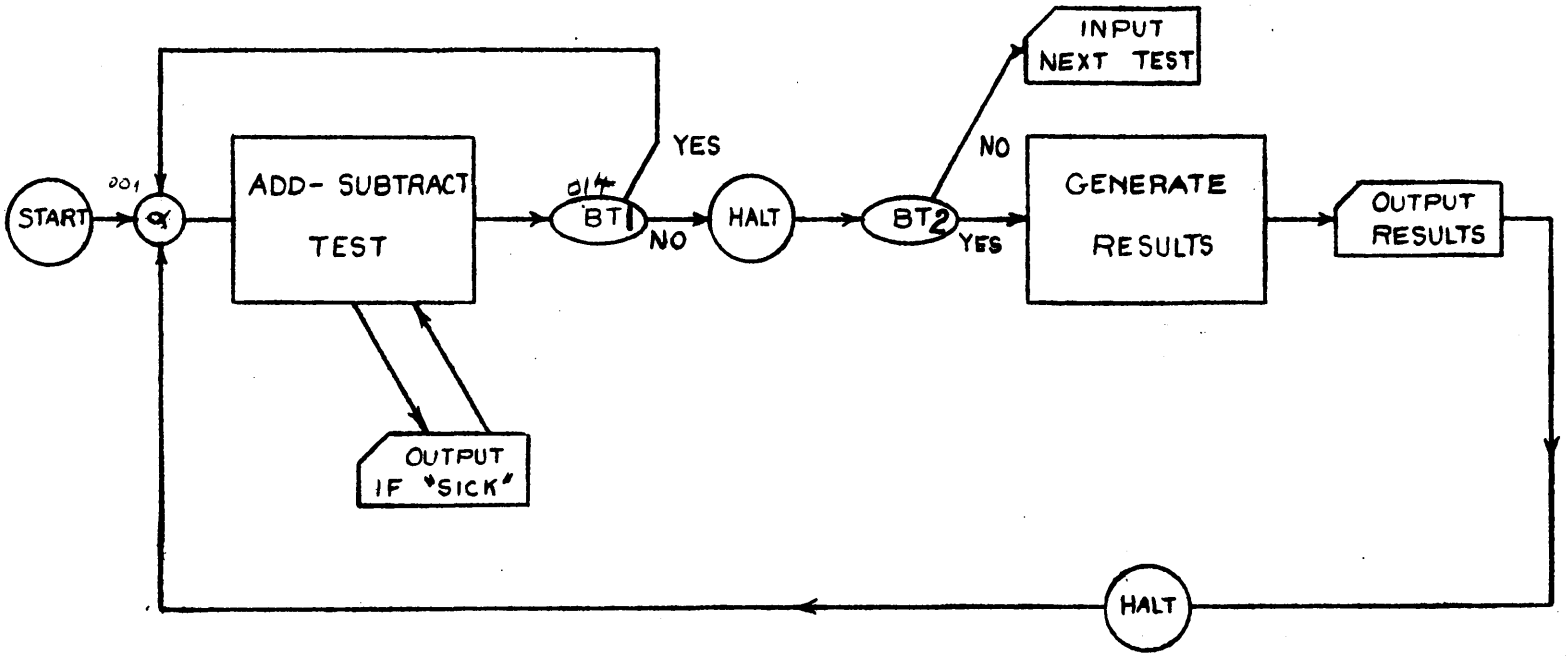


FIGURE 25
FLOW DIAGRAM OF DIAGNOSTIC TEST
ROUTINE 5 (DTR-5)

WISCoding for DIAGNOSTIC TEST ROUTINE 5 (DTR5)

By A. K. Scidmore Date _____ Page 1 of 3

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
1.00	1		A	[41]	[1023]	[35]	001	017	8	029	3ff	023
1.10	2		A	[1023]	[41]	[36]	2	018	8	3ff	029	024
1.20	3		A	[41]	[42]	[37]	3	019	8	029	02a	025
1.30	4		A	[43]	[41]	[38]	4	020	8	02b	029	026
1.40	5		5	[1023]	[41]	[39]	5	021	a	3ff	029	027
2.00	6		CZ	[41]	[1023]	[24]	6	013	c	029	3ff	018
2.10	7		CZ	[1023]	[41]	[25]	7	014	c	3ff	029	019
2.20	8		CN	[41]	[41]	[26]	008	015	e	029	029	01a
2.30	9		CN	[42]	[42]	[27]	9	016	e	02a	02a	01b
2.40	10		CZ	[41]	[35]	[12]	a	017	c	029	023	00c
2.41	11		O	[49]	[49]	[12]	c05b	017	7	031	031	00c
2.50	12		CZ	[41]	[36]	[14]	c	018	c	029	024	00e
2.51	13		O	[50]	[50]	[14]	d	018	7	032	032	00e
2.60	14		CZA	[1023]	[37]	[16]	e	019	d	3ff	025	010
2.61	15		O	[51]	[51]	[16]	f	019	7	033	033	010
2.70	16		CZ	[38]	[42]	[18]	010	020	c	026	02a	012
2.71	17		O	[52]	[52]	[18]	1	020	7	034	034	012
2.80	18		CZA	[41]	[39]	[20]	2	021	d	029	027	014
2.81	19		O	[53]	[53]	[20]	3	021	7	035	035	014
3.00	20		BT	[—]	[—]	[1]	4	100	5	000	000	001
3.10	21		H	[—]	[—]	[22]	5	000	6	000	000	016
3.20	22		BT	[—]	[—]	[28]	6	100	5	000	000	01c
3.30	23		I	[—]	[—]	[—]	7	000	0	000	000	000
2.01	24		O	[45]	[45]	[7]	018	013	7	02d	02d	007
2.11	25		O	[46]	[46]	[8]	019	014	7	02e	02e	008

WISCoding for DIAGNOSTIC TEST ROUTINE 5 (DTR5)

By _____ Date _____ Page 2 of 3

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
2.21	26		0	[47]	[47]	[9]	003a	015	7	02f	02f	009
2.31	27		0	[48]	[48]	[10]	004b,b	016	7	030	030	00a
4.00	28		S	[41]	[41]	[33]	01C	015	a	029	029	021
4.10	29		S	[42]	[42]	[34]	01d	016	a	02a	02a	022
4.20	30		SA	[41]	[42]	[40]	01C	021	b	029	02a	028
4.30	31		0	[33]	[44]	[32]	01f	000	7	021	02c	020
4.40	32		H	[/]	[/]	[1]	020	000	6			001
	33			[ ]	[ ]	[ ]	1	000	0	000	000	000
	34			[ ]	[ ]	[ ]	2	000	0	000	000	000
	35			[ ] OPSTO [FOR		]	3	000	1	234	567	89a
	36			[ ] RESULTS [		]	4	000	1	234	567	89a
	37			[ ]	[ ]	[ ]	5	000	0	000	000	000
	38			[ ]	[ ]	[ ]	6	200	1	234	567	89a
	39			[ ]	[ ]	[ ]	7	200	1	234	567	89a
	40			[ ]	[ ]	[ ]	028	000	0	000	000	000
	41			[ ]	[ ]	[ ]	9	000	1	234	567	89a
	42			[ ] CONSTANTS		]	A	200	1	234	567	89a
	43			[ ]	[ ]	[ ]	b	200	2	468	ace	134
	44			[ ]	[ ]	[ ]	C	200	0	000	000	000
	45			[ ]	[ ]	[ ]	d	123	4	e01	e01	e01
	46			[ ]	[ ]	[ ]	E	123	4	e02	e02	e02
	47			[ ] KEYS	[	]	f	123	4	e03	e03	e03
	48			[ ]	[ ]	[ ]	030	123	4	e04	e04	e04
	49			[ ]	[ ]	[ ]	1	123	4	e05	e05	e05
	50			[ ]	[ ]	[ ]	2	123	4	e06	e06	e06

WISCoding for DIAGNOSTIC TEST ROUTINE 5 (DTR5)

By ______________ Date ______________ Page 3 of 3

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	51			[ ]	[ ]	[ ]		123	4	e07	e07	e07
	52			[ ]	KEYS [ ]	[ ]		123	4	e08	e08	e08
	53			[ ]	[ ]	[ ]		123	4	e09	e09	e09
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

fault can be determined from a cause and effect table (Figure 26) similar to that prepared for DTR4.

Almost all of the information which can be gleaned from an examination of the answers to be punched out can also be deduced from the patterns of key constants. However, certain types of faults will give the same pattern of keys; these can be distinguished only with the aid of the results of the test orders. For example, there is no reliable compare test to determine whether the AA operand loop is "dropping" bits, or whether the BB operand loop is "adding" bits, and these two faults will be listed together in the cause and effect table. In this case, then, the results of orders one and two will have to be consulted to separate them.

Faults of the type listed in Figure 26 are what the author terms "dead" faults; i.e., the signal in question is either never present or always present, always "0" or always "1". Not only is a dead fault somewhat analogous to a "dead short", but they are both frequently traceable to vacuum tube heater "burn-out"--a dead tube. The other type of fault which can occur is called an intermittent fault since it may or may not be present at one particular instant. An intermittent fault is almost always the more difficult of the two to diagnose.

The cause and effect table shown for the routine (Figure 26) is obviously on a very elementary level. Since there are several hundred different circuits involved in this test, and since each circuit may fail in more than one manner, the very large amount of labor which would be required to compile a complete table showing all possible faults and the pattern of key constants each would produce is neither necessary nor practical, and a table which is much simpler, yet a great deal more

comprehensive than Figure 26, will suffice. Such a table, although not available at this time, is being prepared to satisfy the immediate needs. However, when the floating point unit is added to the WISC both this diagnostic routine and its cause and effect table will have to be revised.

f. The Final Test - DTR6

If the computer is known to contain a fault, but has successfully passed routines DTR2, 3, 4 and 5, the fault must exist in the circuits specifically associated with the multiply and divide orders, the extract order, short memory and read-write coincidence, or the tracks not used in the performance of the above routines. The object of Diagnostic Test Routine 6 (DTR6), pages 130 - 136, is to check each of the above and indicate wherein the error lies.

As can be seen from the flow diagram, Figure 27, the routine is divided into four parts, each testing one of the aforementioned groups of circuits. Should there be no indication of failure during the performance of any of the first three tests, the loading instruction for the next sequential test can be reached by means of the breakpoint transfer order. The track test portion of this routine which causes reading and writing of orders and operands in all of the available memory locations not previously used by the other diagnostic routines is very similar to the Acceptance Test Routine in that the test is relocated in the drum each time it is completed successfully.

The location of faults from an examination of the key constants for this routine is in general not difficult since each constant indicates a malfunction in a particular circuit or track. Hence,

FIGURE 26

CAUSE AND EFFECT TABLE FOR DTR5

KEY	CAUSE												
	1	2	3	4	5	6	7	8	9	10	11	12	13
e01	X	X											
e02		X	X										
e03	X			X									
e04			X		X								
e05			X	X	X	X	X	X					
e06	X		X	X	X	X	X	X	X	X			
e07	X			X	X			X	X	X	X		
e08	X			X	X		X	X	X	X	X	X	
e09	X		X	X	X		X	X		X		X	X

DEAD FAULTS:

1. Output of AA Loop = "0"
2. Output of RR Loop = "0"
3. Output of either BB or B Loop = 0
4. Output of BB Loop = 1
5. Output of RR Loop = 1
6. "RS" signal always present (Result sign always positive)
7. "SC" signal never present
8. Output of B Loop = 1
9. "SC" signal always present
10. Output of AA Loop = 1
11. "Alg" signal never present
12. "RS" signal never present
13. "Alg" signal always present

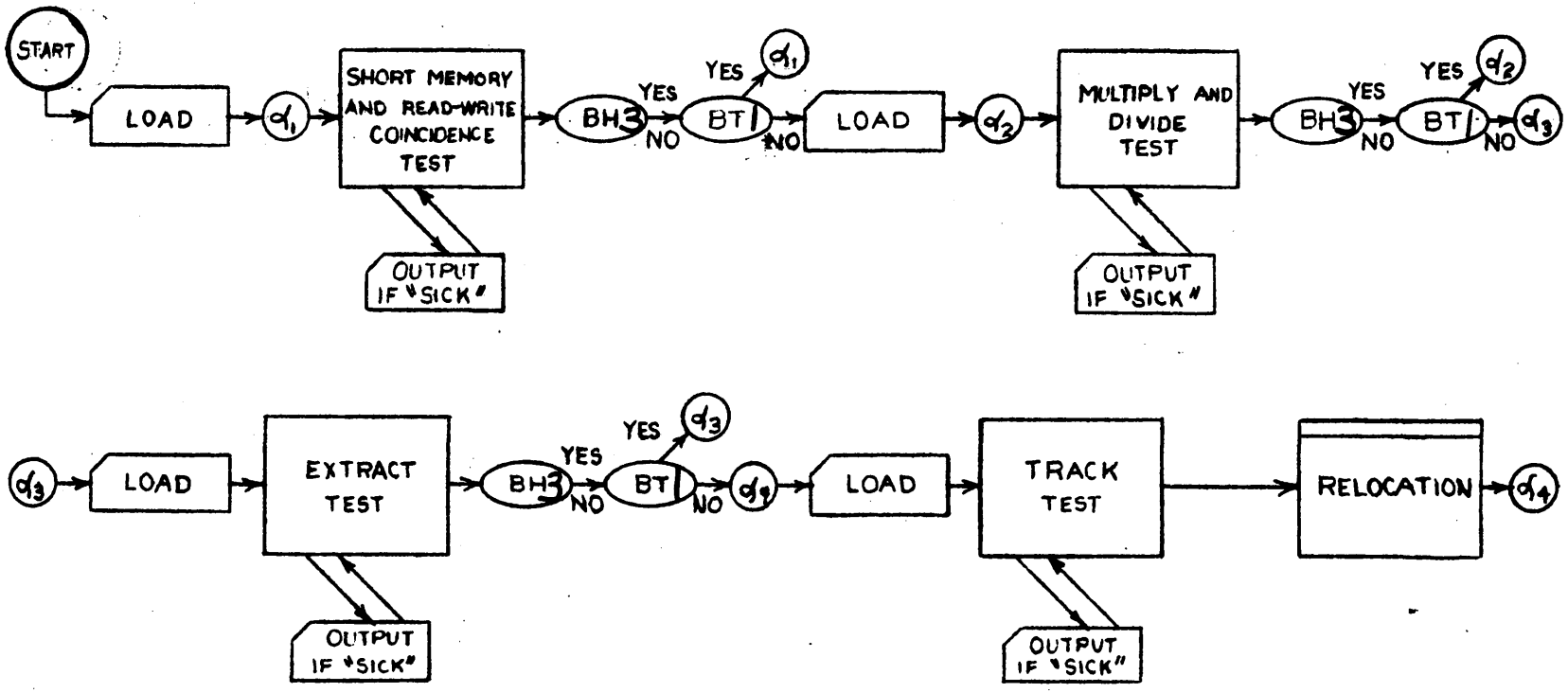


FIGURE 27
FLOW DIAGRAM OF DIAGNOSTIC TEST
ROUTINE 6 (DTR-6)

fo1 - Comm: Result & order

fo2 - SMA

fo3 - comm: Result & order

fo4 - SMB

fo5 - Comm Result & order B

fo6 - SM result

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By A. K. Scidmore Date _____ Page 1 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	0		I	[1]	[29]	[1]		000	0	001	01d	001
	1		A	[23]	[1023]	[36]	001	000	8	017	3ff	024
	2		A	[ ]	[1023]	[30]	2	000	8	800	3ff	01e
	3		A	[23]	[1023]	[ ]	3	000	8	017	3ff	800
	4		TZ	f01 [36]	[23]	[6]	4	000	c	024	017	006
	5		O	[24]	[24]	[6]	5	000	7	018	018	006
	6		TZ	f06 [0]	[23]	[19]	6	000	c	000	017	013
	7		TZ	f02 [30]	[23]	[9]	7	000	c	01e	017	009
	8		O	[25]	[25]	[9]	8	000	7	019	019	009
	9		A	[1023]	[23]	[43]	9	000	8	3ff	017	02b
	10		A	[1023]	[ ]	[30]	a	000	8	3ff	800	01e
	11		TZ	f03 [43]	[23]	[13]	b	000	c	02b	017	00d
	12		O	[26]	[26]	[13]	c	000	7	01a	01a	00d
	13		TZ	f04 [23]	[30]	[15]	d	000	c	017	01e	00f
	14		O	[27]	[27]	[15]	e	000	7	01b	01b	00f
	15		A	[23]	[1023]	[49]	00f	000	8	017	3ff	031
	16		A	[23]	[1023]	[ ]	010	000	8	017	3ff	800
	17		TZ	f05 [~~49~~ 23]	[~~23~~ 49]	[20]	1	000	c	~~033~~ 017	~~017~~ 031	014
	18		O	[28]	[~~29~~ 28]	[20]	2	000	7	01c	~~01d~~ 018	014
	19		O	[~~25~~ 39]	[~~25~~ 29]	[7]	3	000	7	~~019~~ 01d	019 01a	007
	20		BH	[ ]	[ ]	[21]	014	100	6	000	000	015
	21		BT	[ ]	[ ]	[1]	15	100	5	000	000	001
	22		I	[ ]	[ ]	[ ]	16	000	0	000	000	000
	23			CONSTANT C	[ ]	[ ]	17	000	5	6ef	656	efe
	24		}	KEYS	[ ]	[ ]	018	123	4	f01	f01	f01

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By _____ Date _____ Page 2 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	25			[ ]	[ ]	[ ]	019	123	4	f02	f02	f02
	26			[ ]	[ ]	[ ]		123	4	f03	f03	f03
	27			[KEY$	[ ]	[ ]		123	4	f04	f04	f04
	28			[ ]	[ ]	[ ]		123	4	f05	f05	f05
	29			[ ]	[ ]	[ ]	01d	123	4	f06	f06	f06
	30			[ ]	[ ]	[ ]	01e	000	5	6ef	e56	efe
	36			[ ]	[ ]	[ ]	024	000	5	6ef	e56	efe
	43			OPSTOS [ ]	[ ]	[ ]	026	"	"	"	"	"
	49			[ ]	[ ]	[ ]	031	"	"	"	"	"
	0		I	[1]	[29]	[1]		000	0	001	01d	001
	1		M	C_1[~~39~~ 19]×C_2 [~~40~~ 20]=C_3 [30]			1	000	2	~~027~~ 013	~~028~~ 014	01e
	2		M	C_2[~~40~~ 20]×C_1 [~~39~~ 19]=C_3 [31]			2	000	2	~~028~~ 014	~~027~~ 013	01f
	3		M	C_1[~~39~~ 19]×C_1 [~~39~~ 19]=C_4 [32]			3	000	2	~~027~~ 013	~~027~~ 013	020
	4		M	C_2[~~40~~ 20]×C_2 [~~40~~ 20]=C_5 [33]			4	000	2	~~028~~ 014	~~028~~ 014	021
	5		D	C_5[~~43~~ 800]÷C_2 [~~40~~ 20]=C_6 [34]			5	000	3	~~026~~ 800	~~028~~ 014	022
	6		TZ	C_3[30]-C_3 [21]=0 [8]			6	000	c	01e	015	008
	7		0	C [25] [25] [8]			7	000	7	019	019	008
	8		TZ	C_3[31]-C_3 [21]=0 [10]			8	000	c	01f	015	00a
	9		0	[26] [26] [10]			9	000	7	01a	01a	00a
	10		TZ	C_4[32]-C_4 [22]=0 [12]			a	000	c	020	016	00c
	11		0	[27] [27] [12]			b	000	7	01b	01b	00c
	12		TZ	C_5[33]-C_5 [23]=0 [14]			c	000	c	021	017	00e
	13		0	[28] [28] [14]			d	000	7	01c	01c	00e
	14		TZ	C_6[34]-C_6 [24]=0 [16]			e	000	c	022	018	010
	15		0	[29] [29] [16]			0f	000	7	01d	01d	010

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By _____ Date _____ Page 3 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	16		BH	[/]	[/]	[17]	018	100	6	000	000	011
	17		BT	[/]	[/]	[1]	1	100	5	000	000	001
	18		I	[/]	[/]	[/]	2	000	0	000	000	000
	19			C_1 [ ]	[ ]	[ ]	013	000	f	ea8	6dd	b53
	20			C_2 [ ]	[ ]	[ ]	014	200	f	aca	de6	6d5
	21			C_3 [ ]	[ ]	[ ]	015	200	f	97a	428	fae
	22			C_4 [CONSTANTS	[ ]	[ ]	016	000	f	d52	9ae	265
	23			C_5 [ ]	[ ]	[ ]	017	000	f	560	db2	493
	24			C_6 [ ]	[ ]	[ ]	018	200	f	aca	de6	6d4
	25			f_{07} [ ]	[ ]	[ ]	019	123	4	f07	f07	f07
	26			f_{08} [ ]	[ ]	[ ]	01a	123	4	f08	f08	f08
	27			f_{09} [KEYS	[ ]	[ ]	b	123	4	f09	f09	f09
	28			f_{10} [ ]	[ ]	[ ]	c	123	4	f10	f10	f10
	29			f_{11} [ ]	[ ]	[ ]	d	123	4	f11	f11	f11
	30			[ ]	[ ]	[ ]	e	200	f	97a	428	fae
	31			[ ]	[ ]	[ ]	01f	200	f	97a	428	fae
	32			[OPSTOS	[ ]	[ ]	020	000	f	d52	9ae	265
	33			[ ]	[ ]	[ ]	021	000	f	560	db2	493
	34			[ ]	[ ]	[ ]	022	200	f	aca	de6	6d4
	0		I	O[1]+O	[40]=O	[1]		000	0	001	028	001
	1		A	O[1023]+O	[1023]=O	[41]		000	8	3ff	3ff	029
	2		A	O[1023]+O	[1023]=O	[42]		000	8	3ff	3ff	02a
	3		A	O[1023]+O	[1023]=O	[43]		000	8	3ff	3ff	02b
	4		A	O[1023]+O	[1023]=O	[44]		000	8	3ff	3ff	02c
	5		A	O[1023]+O	[1023]=O	[45]		000	8	3ff	3ff	02d

	Shift	X≠Y	EVEN-ODD	SM	
f12	Left	X<Y	1		11, 10, 39
f13	Left	X<Y	1	SM A	1, 10, 39
f14	NONE	X=Y	/	SM A	1, 1, 50
f15	RIGHT	X>Y	6	/	14, 2, 36
f16	Left	X<Y	1	/	1, 3, 47
f17	Left	X<Y	1	1	1, 4, 46

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DIR6)

By _____ Date _____ Page 4 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	6		A	O [1023] + O	[1023] = O	[46]	6	000	8	3ff	3ff	02e
	7	11	E	C_0[30]	[10,39] = C_1	[41]	7	00b	1	01e	8a7	029
	8		A	O [1023] + C_0	[30] = C_0	[\ ]	8	000	8	3ff	01e	800
	9	11	E	" [\ ]	[10,39] → C_1	[42]	9	00b	1	800	8a7	02a
	10		A	O [1023] + C_0	[30] = C_0	[\ ]	a	000	8	3ff	01e	800
	11	1	E	" [\ ]	[1,50] → C_0	[43]	00b	001	1	800	c12	02b
	12	14	E	C_0[30]	[2,36] → C_2	[44]	c	00e	1	01e	824	02c
	13	1	E	C_0[30]	[3,47] → C_3	[45]	d	001	1	01e	83f	02d
	14	1	E	C_0[30]	[4,46] → C_4	[46]	e	001	1	01e	83e	02e
	15		TZ	C_1[41] - C_1	[31] = O	[17]	00f	000	c	029	01f	011
	16		O	[35]	[35]	[17]	010	000	7	023	023	011
	17		TZ	C_1[~~41~~ 42] - C_1	[31] = O	[19]	1	000	c	~~029~~ 02a	01f	013
	18		O	[36]	[36]	[19]	2	000	7	024	024	013
	19		TZ	C_0[43] - C_0	[30] = O	[21]	3	000	c	02b	01e	015
	20		O	[37]	[37]	[21]	4	000	7	025	025	015
	21		TZ	C_2[44] - C_2	[32] = O	[23]	5	000	c	02c	020	017
	22		O	[38]	[38]	[23]	6	000	7	026	026	017
	23		TZ	C_3[45] - C_3	[33] = O	[25]	7	000	c	02d	021	019
	24		O	[39]	[39]	[25]	8	000	7	027	027	019
	25		TZ	C_4[46] - C_4	[34] = O	[27]	9	000	c	02e	022	01b
	26		O	[40]	[40]	[27]	a	000	7	028	028	01b
	27		BH	[\ ]	[\ ]	[28]	b	100	6	000	000	01c
	28		BT	[\ ]	[\ ]	[1]	c	100	5	000	000	001
	29		I	[\ ]	[\ ]	[\ ]	d	000	0	000	000	000
	30			C_0[CONSTANTS]	[ ]	[ ]	01e	384	f	ca8	66d	b55

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By _____ Date _____ Page 5 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	31			C_1 [ ]	[ ]	[ ]	01f	1c2	7	f54	336	c00
	32			C_2 [ ]	[ ]	[ ]	020	000	1	84f	ea8	66c
	33			C_3 [CONSTANTS]	[ ]	[ ]	1	013	f	aa1	9bc	d54
	34			C_4 [ ]	[ ]	[ ]	2	027	f	543	36d	aa8
	35			f_{12} [ ]	[ ]	[ ]	3	123	4	f12	f12	f12
	36			f_{13} [ ]	[ ]	[ ]	4	123	4	f13	f13	f13
	37			f_{14} [KEYS]	[ ]	[ ]	5	123		f14	f14	f14
	38			f_{15} [ ]	[ ]	[ ]	6	123		f15	f15	f15
	39			f_{16} [ ]	[ ]	[ ]	7	123		f16	f16	f16
	40			f_{17} [ ]	[ ]	[ ]	028	123	4	f17	f17	f17
	41			[ ]	[ ]	[ ]	029	0c2	7	f54	336	c00
	42			[ ]	[ ]	[ ]	02a	000	7	f54	336	c00
	43			[OPSTOS]	[ ]	[ ]	b	200	f	ea8	66d	b55
	44			[ ]	[ ]	[ ]	c	000	1	84f	ea8	66c
	45			[ ]	[ ]	[ ]	d	013	f	aa1	9b6	d54
	46			[ ]	[ ]	[ ]	02c	027	f	543	36d	aa8
	0		I	[1]	[50]	[42]		000	0	001	032	02a
	1	1	E	χ_i [34]	[5,12]	[3]		001	1	022	19c	003
	2	1	E	$\chi_i+\Delta$ [35]	[5,12]	[19]		001	1	023	19c	013
	3	1	E	W_i [x_1]	[1,50]	[32]		001	1	000	c12	020
	4	1	E	W_i [32]	[1,12] → γ	[33]		001	1	020	01c	021
	5		TN	χ_n [36] − γ	[33]	[9]		000	e	024	021	009
	6		TNA	j [38] − "	[ ]	[9]		000	f	026	800	009
	7		A	γ [33] + Δ	[37]	[ ]		000	8	021	025	800
	8	1	E	$\gamma+\Delta$ [ ]	[1,12]	[32]		001	1	800	01c	020

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By _____ Date _____ Page 6 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	9	13	E	W_i[32]	[1,12]→β	[33]		00d	1	020	01c	021
	10		TN	χ_n[36]-β	[33]	[14]		000	e	024	021	00e
	11		TNA	j [38]-"	[ ]	[14]		000	f	026	800	00e
	12		A	β [33]+Δ	[37]	[ ]		000	8	021	025	800
	13	1	E	" [ ]	[13,12]	[32]		001	1	800	0dc	020
	14	25	E	W_i[32]	[1,12]→α	[33]		019	1	020	01c	021
	15		TN	χ_n[36]-α	[33]	[19]		000	e	024	021	013
	16		TNA	j [38]-"	[ ]	[19]		000	f	026	800	013
	17		A	α [33]+Δ	[37]	[ ]		000	8	021	025	800
	18	1	E	α+Δ [ ]	[25,12]	[32]		001	1	800	19c	020
	19	1	E	W_i[32]	[1,50]	[x_1+Δ]		001	1	020	c12	000
	20		A	χ_i[34]+001	[1021] = χ_i	[34]		0008		022	3fd	022
	21		A	" [ ]+Δ	[37] = χ_i+Δ	[35]		000	8	800	025	023
	22		TZ	χ_n[36]-χ_i	[34]	[24]		000	c	024	022	018
	23		TU	[ ]	[ ]	[1]		000	5	000	000	001
	24		S	χ_i+Δ[35]-j	[38] = new χ_i	[34]		000	a	023	026	022
	25		A	χ_i+Δ[35]+Δ	[37] = new	[35]	χ_i+Δ	000	8	023	025	023
	26		A	χ_i [34]+j	[38] = new χ_n	[36]		000	8	022	026	024
	27	1	E	χ_i [34]	[1,12]	[29]		001	1	022	010	027
	28		TN	χ'_n [40]-χ_n	[36]	[30]		000	e	028	024	01e
	29		TU	[ ]	[ ]	[x_1]		000	5	000	000	000
	30		S	χ_i+Δ[35]-Δχ	[39]	[35]		000	a	023	027	023
	31		TU	[ ]	[ ]	[29]		000	5	000	000	01d
	32			[ ] W_i	[ ]	[ ]		000	0	000	000	000
	33			OPSTOS [ ]α,β,γ	[ ]	[ ]		000	0	000	000	000

WISCoding for DIAGNOSTIC TEST ROUTINE 6 (DTR6)

By _____ Date _____ Page 7 of 7

FLOW #	ORDER #	X	TYPE	A	B	C	#	HEXADECIMAL X	T	A	B	C
	34			X_i = [42]	[ ]	[ ]		000	0	000	000	02a
	35			$X_i+\Delta$ = [47]	[OPSTOS]	[ ]		000	0	000	000	04a
	36			X_n = [51]	[ ]	[ ]		000	0	000	000	033
	37			Δ = [32]	[ ]	[ ]		000	0	000	000	020
	38			j = [9]	[CONSTANTS]	[ ]		000	0	000	000	009
	39			ΔX = [915]	[ ]	[ ]		000	0	000	000	393
	40			X_n' = [965]	[ ]	[ ]		000	0	000	000	3c5
	41			C [ ]	[ ]	[ ]						
	42		A	C[41]+O	[1023] = C	[46]		000	8	029	3ff	02e
	43		A	O[1023]+C	[41] = C	[49]		000	8	3ff	029	031
	44		CZ	C[41]-C	[46] = O	[47]		000	c	029	02e	02f
	45		O	[42]	[50]	[47]		000	7	02a	032	02f
	46			[OPSTD]	[ ]	[ ]		000	0	000	000	000
	47		CZ	C[49]-C	[41] = O	[1]		000	c	031	029	001
	48		O	[42]	[50]	[50]		000	7	02a	032	032
	49			[OPSTD]	[ ]	[ ]		000	0	000	000	000
	50		H	[50]	[50]	[1]		000	6	032	032	001
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						
				[ ]	[ ]	[ ]						

the malfunction has already been localized as far as is possible with a test routine. However, as in the case of DTR5, a cause and effect table is presently being prepared to assist in the location of faults. See back of p 129

BIBLIOGRAPHY

1. Asmuth, J. L., C. H. Davidson, J. B. Miller, D. S. Noble, and A. K. Scidmore, "The Wisconsin Integrally Synchronized Computer--A University Research Project", Communications and Electronics, 25 (July 1956), 330-338.

2. Astrahan, M. M. and L. R. Walters, "Reliability of an Air Defense Computing System: Marginal Checking and Maintenance Programming", IRE Transactions on Electronic Computers, EC-5, (December 1956), 233-237.

3. Clement, O. R., "A Basic Routine Library and Programming Manual for the WISC", M.S. Thesis, University of Wisconsin, 1957.

4. Dagget, Norman L., and Edwin S. Rich, "Diagnostic Programs and Marginal Checking in the Whirlwind I Computer", Convention Record of the IRE (Part 7) 1953, 48-54.

5. Davidson, Charles H., "Mathematical Procedures and Testing Methods For an Intermediate Speed Digital Computer", Ph.D. Thesis, University of Wisconsin, 1952.

6. Eachus, J. J., "Group Discussion on Diagnostic Checks", Proceedings of the Eastern Joint Computer Conference, 1953, 119.

7. Eckert, J. Presper, Jr., "Checking Circuits and Diagnostic Routines", Convention Record of the IRE, 1953, 62-65.

8. Goetz, J. A., and H. J. Geisler, "Electron Tube and Crystal Diode Experience in Computing Equipment", Proceedings of the Eastern Joint Computer Conference, 1953, 61-72.

9. Haraldsen, Hans P., "Extract Order Circuit for an Electronic Digital Computer", M.S. Thesis, University of Wisconsin, 1956.

10. Heath, Harold F.,Jr., "Reliability of an Air Defense Computing System: Component Development", IRE Transactions on Electronic Computers, EC-5, (December 1956), 224-226.

11. Mauchly, John W., "The Advantages of Built-in Checking", Proceedings of the Eastern Joint Computer Conference, 1953, 99-101.

12. Nienburg, Raymond E., "Reliability of an Air Defense Computing System: Circuit Design", IRE Transactions on Electronic Computers, EC-5 (December 1956), 227-233.

13. Noble, David S., "The Design of an Order Interpreter for an Intermediate Speed Digital Computer", Ph.D. Thesis, University of Wisconsin, 1954.

14. Shupe, P. D. Jr., and R. A. Kirsch, "SEAC - Review of Three Years of Operation", Proceedings of the Eastern Joint Computer Conference, 1953, 83-90.

15. Taylor, Norman H., "Marginal Checking as an Aid to Computer Reliability", Proceedings of the IRE, 38-12 (1950), 1418-1421.

16. Wilkes, Maurice V., Montgomery Philster Junr., and Sidney A. Barton, "Experience with Marginal Checking and Automatic Routining of the EDSAC", Convention Record of the IRE, 1953, 66-71.